

(19) World Intellectual Property Organization  
International Bureau



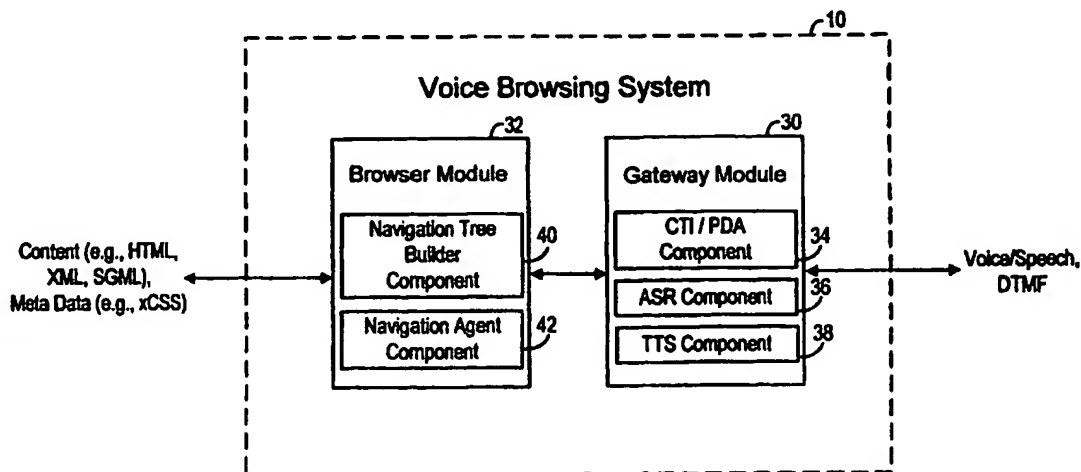
(43) International Publication Date  
17 May 2001 (17.05.2001)

PCT

(10) International Publication Number  
**WO 01/35235 A1**

- (51) International Patent Classification<sup>7</sup>: G06F 13/38, 15/17
- (21) International Application Number: PCT/US00/30749
- (22) International Filing Date:  
8 November 2000 (08.11.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
60/164,429 9 November 1999 (09.11.1999) US  
09/614,504 11 July 2000 (11.07.2000) US
- (71) Applicant: VOCAL POINT, INC. [US/US]; 847 Howard Street, San Francisco, CA 94103 (US).
- (72) Inventors: CHINN, Garry; 850 Campus Drive #214, San Francisco, CA 94015 (US). DUGAN, Benedict, R.; 375 Hermann Street, San Francisco, CA 94131 (US). HAGEN, Roger, E.; 3426 21st Street #A, San Francisco, CA 94110 (US). SEXTON, Michael, R.; 3830 18th Street, San Francisco, CA 94114 (US).
- (74) Agents: WOO, Philip, W. et al.; Skjerven, Morrill, MacPherson LLP., 25 Metro Drive, Suite 700, San Jose, CA 95110 (US).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:**  
— With international search report.
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

(54) Title: SYSTEM AND METHOD FOR ACCESSING WEB CONTENT USING LIMITED DISPLAY DEVICES



(57) Abstract: A computer system is provided for allowing a user of a limited display device (34) to browse content available from a data network. The system includes an interface (32) which receives a request for the content from the user via the limited display device. A processor, coupled to the interface, retrieves a conventional markup language document containing the content from the data network. The processor converts the conventional markup language document into a navigation tree (40) which provides a semantic, hierarchical structure for the content.



WO 01/35235 A1

SYSTEM AND METHOD FOR ACCESSING WEB CONTENT  
USING LIMITED DISPLAY DEVICES

5 TECHNICAL FIELD OF THE INVENTION

The present invention relates generally to data networks and, in particular, to a system and method for accessing web content using limited display devices.

10 BACKGROUND OF THE INVENTION

The Internet, and the closely related application known as the "World Wide Web," have made an abundance of information (e.g., public news, stock quotes, product information, etc.) readily available to anyone with a desktop computer running a  
15 conventional web browser program, such as Microsoft's Internet Explorer or Netscape's Communicator. Such a web browser provides an interface between the Internet or other data networks and the desktop computer, allowing a desktop computer user to view information at one or more web pages. Web pages are supported by  
20 documents formatted in a conventional markup language such as Hyper-Text Markup Language (HTML) and eXtensible Markup Language (XML). Although these markup languages are suitable for presenting information on a desktop computer, they are generally not well suited for new, emerging devices--such as, for example,  
25 cellular telephones, smart telephones, wireless personal digital assistants (PDAs), and like devices with limited display capability--through which Internet information could potentially

be delivered. Furthermore, neither conventional web browsers nor conventional markup languages support or allow users to readily access information from the Internet using voice commands or commands from limited display devices.

5        Efforts have been made to address such problems. For example, voice-enabling languages, such as, Voice Extensible Markup Language (VoiceXML) have been developed. Unlike the conventional markup languages of the Internet (e.g., HTML and XML), VoiceXML enables the delivery of information via voice  
10    commands or commands from limited display devices. However, any information which is desirably delivered with VoiceXML must be separately constructed in that language, apart from the conventional markup languages. Because most websites on the Internet do not provide separate VoiceXML capability, much of the  
15    information on the Internet is still largely unavailable to people without desktop computers or via voice commands.

#### SUMMARY OF THE INVENTION

      According to an embodiment of the present invention, a  
20    computer system is provided for allowing a user of a limited display device to browse content available from a data network. The system includes an interface that receives a request for the content from the user via the limited display device. A processor, coupled to the interface, retrieves a conventional  
25    markup language document containing the content from the data network. The processor converts the conventional markup language

document into a navigation tree that provides a semantic, hierarchical structure for the content.

According to another embodiment of the present invention, a method performed on a computer is provided for allowing a user of a limited display device to browse content available from a data network. The method includes the following: receiving a request for the content from the user via the limited display device; retrieving a conventional markup language document containing the content from the data network; and converting the conventional markup language document into a navigation tree which provides a semantic, hierarchical structure for the content. This structure is suitable for presenting content in audible form, and thus, is appropriate for an environment with limited capacity for display.

According to yet another embodiment of the present invention, a computer system for allowing a user of a limited display device to browse content available from a data network includes a markup language parser. The markup language parser receives a conventional markup language document in response to a request for the content from the user via the limited display device. Furthermore, the markup language parser generates a document tree from the conventional markup language document. A style sheet parser receives a style sheet document in response to the request, and generates a style tree from the style sheet document. The style tree comprises a plurality of style sheet rules. A tree converter, which is in communication with the markup language parser and the style sheet parser, converts the document tree into a navigation tree using the style sheet tree

rules. The navigation tree provides a semantic, hierarchical structure for the content.

According to yet another embodiment of the present invention, a method performed on a computer for allowing a user of a limited display device to browse content available from a data network includes: receiving a conventional markup language document and a style sheet document in response to a request for the content from the user via the limited display device; generating a document tree from the conventional markup language document; generating a style tree from the style sheet document, the style tree comprising a plurality of style sheet rules; and converting the document tree into a navigation tree using the style sheet tree rules, the navigation tree providing a semantic, hierarchical structure for the content.

According to yet another embodiment of the present invention, a computer system for allowing a user of a limited display device to browse content available from a data network includes a gateway module. The gateway module is operable to receive a spoken request for the content from the user via the limited display device, and to recognize the spoken request. A browser module, in communication with the gateway module, is operable to retrieve a conventional markup language document and a style sheet document from the data network in response to the spoken request. The conventional markup language document contains the content; the style sheet document contains metadata. The browser module is operable to generate a navigation tree using the conventional markup language document and the style

sheet document. The navigation tree provides a semantic, hierarchical structure for the content. The gateway module and the browser module cooperate to enable the user to browse the content using the navigation tree and to output speech conveying the content to the user via the limited display device.

According to still yet another embodiment of the present invention, a method performed on a computer for allowing a user of a limited display device to browse content available from a data network includes: receiving a spoken request for the content from the user via the limited display device; recognizing the spoken request; retrieving a conventional markup language document and a style sheet document from the data network in response to the spoken request, the conventional markup language document containing the content, the style sheet document containing metadata; generating a navigation tree using the conventional markup language document and the style sheet document, the navigation tree providing a semantic, hierarchical structure for the content; enabling the user to browse the content using the navigation tree; and outputting speech conveying the content to the user via the limited display device.

A technical advantage of the present invention includes providing a system and method for accessing or browsing content available from a data network (e.g., the Internet) using voice commands, for example, from any telephone, wireless personal digital assistant, or other device with limited display capability. This system and method for voice browsing navigates through the content and delivers the same, for example, in the

form of generated speech. The system and method can voice-enable any content currently formatted in a conventional, Internet-accessible markup language (e.g., HTML and XML), thus offering an unparalleled experience for users.

5 To accomplish this, in one embodiment, the system and method of the present invention build or generate navigation trees using the conventional, Internet-accessible markup language documents supporting current websites. A navigation tree organizes the content of a web page into an outline or hierarchical structure  
10 that takes into account the meaning of the content, and thus can be used for semantic retrieval of the content. As such, a navigation tree supports voice-based browsing of web pages by users. For documents formatted in various conventional markup languages, respective default style sheet (e.g., xCSS) documents  
15 may be provided for use in generating the navigation trees. Each style sheet document may contain metadata, such as declarative statements (rules) and procedural statements. For each conventional markup language document, the system and method may construct a document tree comprising a number of nodes. The  
20 rules or declarative statements contained in a suitable style sheet document are used to modify the document tree, for example, by adding or modifying attributes at each node of the document tree, deleting unnecessary nodes, filtering other nodes, etc. If procedural statements are present in the style sheet document,  
25 the system and method may apply these procedures directly to construct the navigation tree. If there are no such procedural

statements, the system and method may apply a simple mapping procedure to convert the document tree into the navigation tree.

Other aspects and advantages of the present invention will become apparent from the following descriptions and accompanying  
5 drawings.

#### BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and for further features and advantages, reference is now made to  
10 the following description taken in conjunction with the accompanying drawings, in which:

FIG. 1A illustrates an exemplary environment in which a voice browsing system, according to an embodiment of the present invention, may operate;

15 FIG. 1B illustrates another exemplary environment in which a voice browsing system, according to an embodiment of the present invention, may operate;

FIG. 2 is a block diagram of a voice browsing system, according to an embodiment of the present invention;

20 FIG. 3 is a block diagram of a navigation tree builder component, according to an embodiment of the present invention;

FIG. 4 is a block diagram of a tree converter, according to an embodiment of the present invention;

FIG. 5 illustrates an exemplary document tree;

25 FIG. 6 illustrates an exemplary navigation tree, according to an embodiment of the present invention;



FIG. 7 illustrates a computer-based system which is an exemplary hardware implementation for the voice browsing system;

FIG. 8 is a flow diagram of an exemplary method for browsing content with voice commands, according to an embodiment of the present invention;

FIG. 9 is a flow diagram of an exemplary method for generating a navigation tree, according to an embodiment of the present invention;

FIG. 10 is a flow diagram of an exemplary method for applying style sheet rules to a document tree, according to an embodiment of the present invention;

FIG. 11 is a flow diagram of an exemplary method for applying heuristic rules to a document tree, according to an embodiment of the present invention; and

FIG. 12 is a flow diagram of an exemplary method for mapping a document tree into a navigation tree, according to an embodiment of the present invention.

#### DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The preferred embodiments of the present invention and their advantages are best understood by referring to FIGS. 1-12 of the drawings. Like numerals are used for like and corresponding parts of the various drawings.

Turning first to the nomenclature of the specification, the detailed description which follows is represented largely in terms of processes and symbolic representations of operations performed by conventional computer components, such as a local or

remote central processing unit (CPU) or processor associated with a general purpose computer system, memory storage devices for the processor, and connected local or remote pixel-oriented display devices. These operations include the manipulation of data bits  
5 by the processor and the maintenance of these bits within data structures resident in one or more of the memory storage devices. Such data structures impose a physical organization upon the collection of data bits stored within computer memory and represent specific electrical or magnetic elements. These  
10 symbolic representations are the means used by those skilled in the art of computer programming and computer construction to most effectively convey teachings and discoveries to others skilled in the art.

For purposes of this discussion, a process, method, routine,  
15 or sub-routine is generally considered to be a sequence of computer-executed steps leading to a desired result. These steps generally require manipulations of physical quantities. Usually, although not necessarily, these quantities take the form of electrical, magnetic, or optical signals capable of being stored,  
20 transferred, combined, compared, or otherwise manipulated. It is conventional for those skilled in the art to refer to these signals as bits, values, elements, symbols, characters, text, terms, numbers, records, files, or the like. It should be kept in mind, however, that these and some other terms should be  
25 associated with appropriate physical quantities for computer operations, and that these terms are merely conventional labels

applied to physical quantities that exist within and during operation of the computer.

It should also be understood that manipulations within the computer are often referred to in terms such as adding, comparing, moving, searching, or the like, which are often associated with manual operations performed by a human operator. It must be understood that no involvement of the human operator may be necessary, or even desirable, in the present invention. The operations described herein are machine operations performed in conjunction with the human operator or user that interacts with the computer or computers.

In addition, it should be understood that the programs, processes, methods, and the like, described herein are but an exemplary implementation of the present invention and are not related, or limited, to any particular computer, apparatus, or computer language. Rather, various types of general purpose computing machines or devices may be used with programs constructed in accordance with the teachings described herein. Similarly, it may prove advantageous to construct a specialized apparatus to perform the method steps described herein by way of dedicated computer systems with hard-wired logic or programs stored in non-volatile memory, such as read-only memory (ROM).

#### Exemplary Environment

FIG. 1A illustrate an exemplary environment in which a voice browsing system 10, according to an embodiment of the present invention, may operate. In this environment, one or more content

providers 12 may provide content to any number of interested users. Each content provider can be an entity which operates or maintains a portal or any other website through which content can be delivered. Each portal or website, which can be supported by  
5 a suitable computer system or web server, may include one or more web pages at which content is made available. Each website or web page can be identified by a respective uniform resource locator (URL).

Content can be any data or information that is presentable  
10 (visually, audibly, or otherwise) to users. Thus, content can include written text, images, graphics, animation, video, music, voice, and the like, or any combination thereof. Content can be stored in digital form, such as, for example, a text file, an image file, an audio file, a video file, etc. This content can  
15 be included in one or more web pages of the respective portal or website maintained by each content provider 12.

These web pages can be supported by documents formatted in a conventional, Internet-accessible markup language, such as, for example, Hyper-Text Markup Language (HTML) and eXtensible Markup  
20 Language (XML). HTML and XML are markup language standards set by the World Wide Web Consortium (W3C) for Internet-accessible documents. In general, conventional markup languages provide formatting and structure for content that is to be presented visually. That is, conventional markup languages describe the  
25 way that content should be displayed, for example, by specifying that text should appear in boldface, which location a particular image should appear, etc. In markup languages, tags are added or

embedded within content to describe how the content should be formatted and displayed. A conventional, Internet-accessible markup language document can be the source page for any browser on a computer.

5        Along with the content, each content provider 12 may also maintain metadata that can be used to guide the construction of a semantic representation for the content. Metadata may include, for example, declarative statements (rules) and procedural statements. This metadata can be contained in one or more style  
10 sheet documents, which are essentially templates that apply formatting and style information to the elements of a web page. A style sheet document can be, for example, an extended Cascading Style Sheet (xCSS) document. In one embodiment, a separate default style sheet documents may be provided for each  
15 conventional markup language (e.g., HTML or XML). As an alternative to style sheets, metadata can be contained in documents formatted in a suitable descriptive language such as Resource Description Framework. Using style sheet documents (or other appropriate documents), auxiliary metadata can be applied  
20 to a web page supported by a conventional markup language document.

One or more data networks, such as the Internet 14, can be used to deliver content. Internet 14 is an interconnection of computer clients and servers located throughout the world and  
25 exchanging information according to Transmission Control Protocol/Internet Protocol (TCP/IP), Internetwork Packet eXchange/Sequence Packet eXchange (IPX/SPX), AppleTalk, or other

suitable protocol. Internet 14 supports the distributed application known as the "World Wide Web." As described herein, web servers maintain websites, each comprising one or more web pages at which information is made available for viewing. Each website or web page may be supported by documents formatted in any suitable conventional markup language (e.g., HTML or XML). Clients may locally execute a conventional web browser program. A conventional web browser is a computer program that allows exchange information with the World Wide Web. Any of a variety of conventional web browsers are available, such as NETSCAPE NAVIGATOR from Netscape Communications Corp., INTERNET EXPLORER from Microsoft Corporation, and others that allow convenient access and navigation of the Internet 14. Information may be communicated from a web server to a client using a suitable protocol, such as, for example, Hypertext Transfer Protocol (HTTP) or File Transfer Protocol (FTP).

A service provider 16 is connected to Internet 14. As used herein, the terms "connected," "coupled," or any variant thereof, mean any connection or coupling, either direct or indirect, between two or more elements; such connection or coupling can be physical or logical. Service provider 16 may operate a computer system that appears as a client on Internet 14 to retrieve content and other information from content providers 12.

In general, service provider 16 can be an entity that delivers services to one or more users. These services may include telephony and voice services, including plain old telephone service (POTS), digital services, cellular service,

wireless service, pager service, etc. To support the delivery of services, service provider 16 may maintain a system for communicating over a suitable communication network, such as, for example, a telecommunications network. Such telecommunications network allows communication via a telecommunications line, such as an analog telephone line, a digital T1 line, a digital T3 line, or an OC3 telephony feed. The telecommunications network may include a public switched telephone network (PSTN) and/or a private system (e.g., cellular system) implemented with a number of switches, wire lines, fiber-optic cable, land-based transmission towers, space-based satellite transponders, etc. In one embodiment, the telecommunications network may include any other suitable communication system, such as a specialized mobile radio (SMR) system. As such, the telecommunications network may support a variety of communications, including, but not limited to, local telephony, toll (i.e., long distance), and wireless (e.g., analog cellular system, digital cellular system, Personal Communication System (PCS), Cellular Digital Packet Data (CDPD), ARDIS, RAM Mobile Data, Metricom Ricochet, paging, and Enhanced Specialized Mobile Radio (ESMR)). The telecommunications network may utilize various calling protocols (e.g., Inband, Integrated Services Digital Network (ISDN) and Signaling System No. 7 (SS7) call protocols) and other suitable protocols (e.g., Enhanced Throughput Cellular (ETC), Enhanced Cellular Control (EC<sup>2</sup>), MNP10, MNP10-EC, Throughput Accelerator (TXCEL), Mobile Data Link Protocol, etc.). Transmissions over the telecommunications

network system may be analog or digital. Transmission may also include one or more infrared links (e.g., IRDA).

One or more limited display devices 18 may be coupled to the network maintained by service provider 16. Each limited display  
5 device 18 may comprise a communication device with limited capability for visual display. Thus, a limited display device 18 can be, for example, a wired telephone, a wireless telephone, a smart phone, a wireless personal digital assistant (PDA), and Internet televisions. Each limited display device 18 supports  
10 communication by a respective user, for example, in the form of speech, voice, or other audible information. Limited display devices 18 may also support dual tone multi-frequency (DTMF) signals.

Voice browsing system 10, as depicted in FIG. 1A, may be  
15 incorporated into a system maintained by service provider 16. Voice browsing system 10 is a computer-based system which generally functions to allow users with limited display devices 18 to browse content provided by one or more content providers 12 using, for example, spoken/voice commands or requests. In  
20 response to these commands or requests, voice browsing system 10, acting as a client, interacts with content providers 12 via Internet 14 to retrieve the desired content. Then, voice browsing system 10 delivers the desired content in the form of audible information to the limited display devices 18. To  
25 accomplish this, in one embodiment, voice browsing system 10 constructs or generates navigation trees using style sheet documents to supply metadata to conventional markup language



(e.g., HTML or XML) documents. Navigation trees are semantic representations of web pages that serve as interactive menu dialogs to support voice-based search by users. Each navigation tree may comprise a number of content nodes and routing nodes.

5 Content nodes contain content from a web page that can be delivered to a user. Routing nodes implement options that can be selected to move to other nodes. For example, routing nodes may provide prompts for directing the user to content at content nodes. Thus, routing nodes link the content of a web page in a  
10 meaningful way. Navigation trees are described in more detail herein.

Voice browsing system 10 thus provides a technical advantage. A voice-based browser is crucial for users having limited display devices 18 since a visual browser is  
15 inappropriate for, or simply cannot work with, such devices. Furthermore, voice browsing system 10 leverages on the existing content infrastructure (i.e., documents formatted in conventional markup languages, such as, HTML or XML) maintained by content providers 12. That is, the existing content infrastructure can  
20 serve as an easy-to-administer, single source for interaction by both complete computer systems (e.g., desktop computer) and limited display devices 18 (e.g., wireless telephones or wireless PDAs). As such, content providers 12 are not required to re-create their content in other formats, deploy new markup  
25 languages (e.g., VoiceXML), or implement additional application programming interfaces (APIs) into their back-end systems to support other formats and markup languages.

Another Exemplary Environment

FIG. 1B illustrates another exemplary environment within which a voice browsing system 10, according to an embodiment of the present invention, can operate. In this environment, voice browsing system 10 may be implemented within the system of a content provider 12. Content provider 12 can be substantially similar to that previously described with reference to FIG. 1A. That is, content provider 12 can be an entity which operates or maintains a portal or any other website through which content can be delivered. Such content can be included in one or more web pages of the respective portal or website maintained by content provider 12. Each web page can be supported by documents formatted in a conventional markup language, such as Hyper-Text Markup Language (HTML) or eXtensible Markup Language (XML). Along with the conventional markup language documents, content provider 12 may also maintain one or more style sheet (e.g., extended Cascading Style Sheet (xCSS)) documents containing metadata that can be used to guide the construction of a semantic representation for the content.

A network 20 is coupled to content provider 12. Network 20 can be any suitable network for communicating data and information. This network can be a telecommunications or other network, as described with reference to FIG. 1A, supporting telephony and voice services, including plain old telephone service (POTS), digital services, cellular service, wireless service, pager service, etc.

A number of limited display devices 18 are coupled to network 20. These limited display devices 18 can be substantially similar to those described with reference to FIG. 1A. That is, each limited display device 18 may comprise a communication device with limited capability for visual display, such as, for example, a wired telephone, a wireless telephone, a smart phone, or a wireless personal digital assistant (PDA). Each limited display device 18 supports communication by a respective user, for example, in the form of speech, voice, or other audible information.

In operation for this environment, voice browsing system 10 again generally functions to allow users with limited display devices 18 to browse content provided by one or more content providers 12 using, for example, spoken/voice commands or requests. In this environment, however, because voice browsing system 10 is incorporated at content provider 12, content provider 12 may directly receive, process, and respond to these spoken/voice commands or requests from users. For each command/request, voice browsing system 10 retrieves the desired content and other information at content provider 12. The content can be in the form of markup language (e.g., HTML or XML) documents, and the other information may include metadata in the form of style sheet (e.g., xCSS) documents. Voice browsing system 10 may construct or generate navigation trees using the style sheet documents to supply metadata to the conventional markup language documents. These navigation trees then serve as interactive menu dialogs to support voice-based search by users.

Voice Browsing System

FIG. 2 is a block diagram of a voice browsing system 10, according to an embodiment of the present invention. In general, voice browsing system 10 allows a user of a limited display device 18 to browse the content available from any one or more content providers 12 using spoken/voice commands or requests. As depicted, voice browsing system 10 includes a gateway module 30 and a browser module 32.

Gateway module 30 generally functions as a gateway to translate data/information between one type of network/computer system and another, thereby acting as an interface. In the context for the present invention, gateway module 30 translates data/information between a network supporting limited display devices 18 (e.g., a telecommunications network) and the computer-based system of voice browsing system 10. For the network supporting the limited display devices, data/information can be in the form of speech or voice.

The functionality of gateway module 30 can be performed by one or more suitable processors, such as a main-frame, a file server, a work station, or other suitable data processing facility supported by memory (either internal or external), running appropriate software, and operating under the control of any suitable operating system (OS), such as MS-DOS, Macintosh OS, Windows NT, Windows 95, OS/2, Unix, Lynix, Xenix, and the like.

Gateway module 30, as shown, comprises a computer telephony interface (CTI)/personal digital assistant (PDA) component 34, an

automated speech recognition (ASR) component 36, and a text-to-speech (TTS) component 38. Each of these components 34, 36, and 38 may comprise one or more programs which, when executed, perform the functionality described herein.

5           CTI/PDA component 34 generally functions to support communication between voice browsing system 10 and limited display devices. CTI/PDA component 34 may comprise one or more application programming interfaces (API) for communicating in any protocol suitable for public switch telephone network (PSTN),  
10   cellular telephone network, smart phones, pager devices, and wireless personal digital assistant (PDA) devices. These protocols may include hypertext transport protocol (HTTP), which supports PDA devices, and PSTN protocol, which supports cellular telephones.

15           Automated speech recognition component 36 generally functions to recognize speech/voice commands and requests issued by users into respective limited display devices 18. Automated speech recognition component 36 may convert the spoken commands/requests into a text format. Automated speech  
20   recognition component 36 can be implemented with automatic speech recognition software commercially available, for example, from the following companies: Nuance Corporation of Menlo Park, CA; Applied Language Technologies, Inc. of Boston, MA; Dragon Systems of Newton, MA; and PureSpeech, Inc. of Cambridge, MA. Such  
25   commercially available software typically can be modified for particular applications, such as a computer telephony application.

Text-to-speech component 36 generally functions to output speech or vocalized messages to users having a limited display device 18. This speech can be generated from content that has been retrieved from a content provider 12 and reformatted within voice browsing system 10, as described herein. Text-to-speech component 38 synthesizes human speech by "speaking" text, such as that which can be part of the content. Software for implementing text-to-speech component 76 is commercially available, for example, from the following companies: Lernout & Hauspie of Ieper, Belgium; AcuVoice, Inc. of San Jose, CA; Centigram Communications Corporation of San Jose, CA; Digital Equipment Corporation (DEC) of Maynard, MA; Lucent Technologies of Murray Hill, NJ; and Entropic Research Laboratory, Inc. of Washington, D.C.

Browser module 32, coupled to gateway module 30, functions to provide access to web pages (of any one or more content providers 12) using Internet protocols and controls navigation of the same. Browser module 32 may organize the content of any web page into a structure that is suitable for browsing by a user using a limited display device 18. Afterwards, browser module 32 allows a user to browse such structure, for example, using voice or speech commands/requests.

The functionality of browser module 32 can be performed by one or more suitable processors, such as a main-frame, a file server, a work station, or other suitable data processing facility supported by memory (either internal or external), running appropriate software, and operating under the control of

any suitable operating system (OS), such as MS-DOS, Macintosh OS, Windows NT, Windows 95, OS/2, Unix, Lynix, Xenix, and the like. Such processors can be the same or separate from that which perform the functionality of gateway module 30.

5       As depicted, browser module 32 comprises a navigation tree builder component 40 and a navigation agent component 42. Each of these components 40 and 42 may comprise one or more programs which, when executed, perform the functionality described herein.

Navigation tree builder component 40 may receive  
10   conventional, Internet-accessible markup language (e.g., XML or HTML) documents and associated style sheet (e.g., xCSS) documents from one or more content providers 12. Using these markup language and style sheet documents, navigation tree builder component 40 generates navigation trees that are semantic  
15   representations of web pages. In general, each navigation tree provides a hierarchical menu by which users can readily navigate the content of a conventional markup language document. Each navigation tree may include a number of nodes, each of which can be either a content node or a routing node. A content node  
20   comprises content that can be delivered to a user. A routing node may implement a prompt for directing the user to other nodes, for example, to obtain the content at a specific content node.

Navigation agent component 42 generally functions to support  
25   the navigation of navigation trees once they have been generated by navigation tree builder component 40. Navigation agent component 42 may act as an interface between browser module 32

and gateway module 30 to coordinate the movement along nodes of a navigation tree in response to any commands and requests received from users.

In exemplary operation, a user may communicate with voice  
5 browsing system 10 to obtain content from content providers 12.  
To do this, the user, via limited display device 18, places a  
call which initiates communication with voice browsing system 10,  
as supported by CTI/PDA component 34 of gateway module 30. The  
user then issues a spoken command or request for content, which  
10 is recognized or interpreted by automatic speech recognition  
component 36. In response to the recognized command/request,  
browser module 32 accesses a web page containing the desired  
content (at a website or portal operated by a content provider  
12) via Internet 14 or other communication network. Browser  
15 module 32 retrieves one or more conventional markup language and  
associated style sheet documents from the content provider.  
Using these markup language and style sheet documents, navigation  
tree builder component 40 creates one or more navigation trees.  
The user may interact with voice browsing system 10, as supported  
20 by navigation agent component 42, to navigate along the nodes of  
the navigation trees. During navigation, gateway module 30 may  
convert the content at various nodes of the navigation trees into  
audible speech that is issued to the user, thereby delivering the  
desired content. Browser module 32 may generate and support the  
25 navigation of additional navigation trees in the event that any  
other command/request from the user invokes another web page of  
the same or a different content provider 12. When a user has



obtained all desired content, the user may terminate the call, for example, by hanging up.

### Navigation Tree Builder Component

5           FIG. 3 is a block diagram of a navigation tree builder component 40, according to an embodiment of the present invention. Navigation tree builder component 40 generally functions to construct navigation trees 50 which can be used to readily and orderly provide the content of respective web pages  
10 to a user via a limited display device 18. As depicted, navigation tree builder 40 comprises a markup language parser 52, a style sheet parser 54, and a tree converter 56. Each of markup language parser 52, style sheet parser 54, and tree converter 56 may comprise one or more programs which, when executed, perform  
15 the functionality described herein.

Markup language parser 52 receives conventional, Internet-accessible markup language (e.g., HTML or XML) documents 58 from a content provider 12. Conventional markup languages describe how content should be structured, formatted, or displayed. To  
20 accomplish this, conventional markup languages may embed tags to specify spans, frames, paragraphs, ordered lists, unordered lists, headlines, tables, table rows, objects, and the like, for organizing content. Each markup language document 58 may serve as the source for a web page. Markup language parser 52 parses  
25 the content contained within a markup language document 58 in order to generate a document tree 60. In particular, markup

language parser 52 can map each markup language document into a respective document tree 60.

Each document tree 60 is a basic data representation of content. An exemplary document tree 60 is illustrated in FIG. 5. Document tree 60 organizes the content of a web page based on, or according to, the formatting tags of a conventional markup language. The document tree is a graphic representation of a HTML document. A typical document tree 60 includes a number of document tree nodes. As depicted, these document tree nodes include an HTML designation (HTML), a header (<HEAD>) and a body (<BODY>), a title (<TITLE>), metadata (<META>), one or more headlines (<H1>, <H2>), lists (<LI>), unordered list (<UL>), a paragraph (<P>). The nodes of a document tree may comprise content and formatting information. For example, each node of the document tree may corresponds to either HTML markup tags or plain text. The content of a markup element appears as its child in the document tree. For example, the header (<HEAD>) may have content in the form of the phrase "About Our Organization" along with formatting information which specifies that the content should be presented as a header on the web page.

Document tree 60 is designed for presenting a number of content elements simultaneously. That is, the organization of web page content according to the formatting tags of conventional markup language documents is appropriate, for example, for a visual display in which textual information can be presented at once in the form of headers, lines, paragraphs, tables, arrays, lists, and the like, along with images, graphics, animation, etc.

However, the structure of a document tree 60 is not particularly well-suited for presenting content serially, for example, as would be required for a audio presentation in which only a single element of content can be presented at a given moment.

5 Specifically, in an audio context, the formatting information of a document tree 60 does not provide meaningful connections or links for the content of a web page. For example, formatting information specifying that content should be displayed as a header does not translate well for an audio presentation of the  
10 content. In addition, much of the formatting information of a document tree 60 does not constitute meaningful content which may be of interest to a user. For example, the nodes for header (<HEAD>) and body (<BODY>) are not intrinsically interesting. In fact, the header (<HEAD>)--comprising title (<TITLE>) and  
15 metadata (<META>)--does not generally contain information that should be presented directly to the user.

Style sheet parser 54 receives one or more style sheet (e.g., xCSS) documents 62. Style sheet documents 62 provide templates for applying style information to the elements of  
20 various web pages supported by respective conventional markup language documents 58. Each style sheet document 62 may supply or provide metadata for the web pages. For example, using the metadata from a style sheet document 62, audio prompts can be added to a standard web page. This metadata can also be used to  
25 guide the construction of a semantic representation of a web page. The metadata may comprise or specify rules which can be applied to a document tree 60. Style sheet parser 54 parses the

metadata from a style sheet document 62 to generate a style tree 64. Each style tree 64 may be associated with a particular document tree 60 according to the association between the respective style sheet documents 62 and conventional markup language documents 58. A style tree 64 organizes the rules (specified in metadata) into a structure by which they can be efficiently applied to a document tree 60. A tree structure for the rules is useful because the application of rules can be a hierarchical process. That is, some rules are logically applied only after other rules have been applied.

Tree converter 56, which is in communication with markup language parser 52 and style sheet parser 54, receives the document trees 60 and style trees 64 therefrom. Using the document trees 60 and style trees 64, tree converter 56 generates navigation trees 50. Among other things, tree converter 56 may apply the rules of a style tree 64 to the nodes of a document tree 60 when generating a navigation tree 50. Furthermore, tree converter 56 may apply other rules (heuristic rules) to each document tree, and thereafter, may map various nodes of the document tree into nodes of a navigation tree 50.

A navigation tree 50 organizes content of a conventional markup language document 58 into a hierarchical or outline structure. With the hierarchical structure, the various elements of content are separated into various levels (e.g., parts, sub-parts, sub-sub-parts etc.). Appropriate mechanisms are provided to allow movement from one level to another and across the levels. The hierarchical arrangement of a navigation tree 50 is

suitable for presenting content sequentially, and thus can be used for "semantic" retrieval of the content at a web page. As such, the navigation tree 50 can serve as an index that is suitable for browsing content using voice commands.

5       An exemplary navigation tree 50 is illustrated in FIG. 6. A navigation tree 50 is, in general, made up of routing nodes and content nodes. Content nodes may comprise content that can be delivered to a user. Content nodes can be of various types, such as, for example, general content nodes, table nodes, and form  
10 nodes. Table nodes present a table of information. Form nodes can be used to assist in the filling out of respective forms. Routing nodes are unique to navigation trees 50 and are generated according to rules applied by tree converter 56. Routing nodes can be used to move between nodes. The routing nodes are  
15 interconnected by directed arcs (edges or links). These directed arcs are used to construct the hierarchical relationship between the various nodes in the navigation tree 50. That is, these arcs specify allowable navigation traversal paths to move from one node to another. In FIG. 6, for example, an unordered list node  
20 (<UL>) is a routing node for moving to list nodes (<L1>, <L2>). The options for other nodes may be explicitly included in the routing node.

Content nodes are not reachable by tree traversal operations. The data found in content nodes must be accessed  
25 through a parent routing node called a group node. The group node organizes content nodes into a single presentational unit. The group node can be used for organizing multi-media content.

For example, rather than present text and links as disjointed content, a group node can be used to organize a collection of text, audio wave files, and URI links together such as the following:

For more information about <A href =  
"http://www.vocalpoint.com/sound.wav"> vocalpoint  
</A>, send email to: <A href = info@vocalpoint.com>  
info@vocalpoint.com </A>.

As such, routing nodes provide the nexus or connection between content nodes, and thus provide meaningful links for the content of a web page. In this way, routing nodes support or provide a semantic, hierarchical relationship for web page content in a navigation tree 50. An exemplary object-oriented implementation for routing and content nodes of a navigation tree is provided in attached Appendix A.

In one embodiment, a navigation tree 50 can be used to define a finite state machine. In particular, various nodes of the navigation tree may correspond to states in the finite state machine. Navigation agent component 42 may use the navigation tree to directly define the finite state machine. The finite state machine can be used by navigation agent 42 of browser module 32 to move throughout the hierarchical structure. At any current state/node, a user can advance to another state/node.

### Tree Converter

FIG. 4 is a block diagram of a tree converter 56, according to an embodiment of the present invention. Tree converter 56

generally functions to convert document trees 60 into navigation trees 50, for example, using style trees 64. As depicted, tree converter 56 comprises a style sheet engine 68, a heuristic engine 70, and a mapping engine 72. Each of style sheet engine 68, heuristic engine 70, and mapping engine 72 may comprise one or more programs which, when executed, perform the functionality described herein.

Style sheet engine 68 generally functions to apply style sheet rules to a document tree 60. Application of style sheet rules can be done on a rule-by-rule basis to all applicable nodes of the document tree 60. These style sheet rules can be part of the metadata of a style sheet document 62. Each style sheet rule can be a rule generally available in a suitable style sheet language of style sheet document 62.

In one embodiment, these style sheet rules may include, for example, clipping, pruning, filtering, and converting. In a clipping operation, a node of a document tree is marked as special so that the node will not be deleted or removed by other operations. Clipping may be performed for content that is important and suitable for audio presentation (e.g., text which can be "read" to a user). In a pruning operation, a node of a document tree is eliminated or removed. Pruning may be performed for content that is not suitable for delivery via speech or audio. This can include visual information (e.g., images or animation) at a web page. Other content that can be pruned may be advertisements and legal disclaimers at each web page. In a filtering operation, auxiliary information is added at a node.

This auxiliary information can be, for example, labels, prompts, etc. In a conversion operation, a node is changed from one type into another type. For example, some content in a conventional markup language document can be in the form of a table for

5 presenting information in a grid-like fashion. In a conversion, such table may be converted into a routing node in a navigation tree to facilitate movement among nodes and to provide options or choices.

As depicted, style sheet engine 68 comprises a selector  
10 module 74 and a rule applicator module 76. In general, selector module 74 functions to select or identify various nodes in a document tree 60 to which the rules may be applied to modify the tree. After various nodes of a particular document tree 60 have been selected by selector module 74, rule applicator module 76  
15 generally functions to apply the various style tree rules (e.g., clipping, pruning, filtering, or converting) to the selected nodes as appropriate in order to modify the tree.

Heuristic engine 70 is in communication with style sheet engine 68. Heuristic engine 70 generally functions to apply one  
20 or more heuristic rules to the document tree 60 as modified by style sheet engine 68. In one embodiment, these heuristic rules may be applied on a node-by-node basis to various nodes of document tree 60. Each heuristic rule comprises a rule which may be applied to a document tree according to a heuristic technique.  
25 A heuristic technique is a problem-solving technique in which the most appropriate solution of several found by alternative methods is selected at successive stages of a problem-solving process for



use in the next step of the process. In the context of the present invention, the problem-solving process involves the process of converting a document tree 60 into a navigation tree 50. In this process, heuristic rules are selectively applied to a document tree after the application of style sheets rules and before a final mapping into navigation tree 50, as described below).

In one embodiment, heuristic rules may include, for example, converting paragraph breaks and line breaks into space breaks (white space), exploiting image alternate tags, deleting decorative nodes, merging content and links, and building outlines from headlines and ordered lists. The operation for converting paragraph breaks and line breaks into space breaks is done to eliminate unnecessary formatting in the textual content at a node while maintaining suitable delineations between elements of text (e.g., words) so that the elements are not concatenated. The operation for exploiting image alternative tags identifies and uses any image alternative tags that may be part of the content contained at a particular node. An image alternative tag is associated with a particular image and points to corresponding text that describes the image. Image alternative tags are generally designed for the convenience of users who are visually impaired so that alternative text is provided for the particular image. The operation for deleting decorative nodes eliminates content that is not useful in a navigation tree 50. For example, a node in the document tree 60 consisting of only an image file may be considered to be a

decorative node since the image itself cannot be presented to a user in the form of speech or audio, and no alternative text is provided. The operation for merging content and links eliminates the formatting for a link (e.g., a hypertext link) is done so  
5 that the text for the link is read continuously as part of the content delivered to a user. The operation for building or generating outlines from headlines and ordered lists is performed to create the hierarchical structure of the navigation tree 50. A headline--which can be, for example, a heading for a section of  
10 a web page--is identified by suitable tags within a conventional markup language document. In a visually displayed web page, multiple headlines may be provided for a user's convenience. These headlines may be considered alternatives or options for the user's attention. An ordered list is a listing of various items,  
15 which in some cases, can be options. Heuristic engine 70 may arrange or organize headlines and ordered lists so that the underlying content is presented in the form of an outline.

Mapping engine 72 is in communication with heuristic engine 70. In general, mapping engine 72 performs a mapping function  
20 that changes certain elements in a modified document tree 60 into appropriate nodes for a navigation tree 50. Mapping engine 72 may operate on a node-by-node basis to provide such mapping function. In one embodiment, the content at a node in document tree 60 is mapped to create a content node in the navigation tree  
25 50. Ordered lists, unordered lists, and table rows are mapped into suitable routing nodes of the navigation tree 50. Any table in document tree 60 may be mapped to create a table node in the

navigation tree 50. A form in a document tree 60 can be mapped to create a form node in the navigation tree 50. A form may comprise a number of fields which can be filled in by a user to collect information. Form elements in the document tree 60 can be mapped into a form handling node in navigation tree 50. Form elements provide a standard interface for collecting input from the user and sending that information to a Web server.

### Computer-Based System

FIG. 7 illustrates a computer-based system 80 which is an exemplary hardware implementation for voice browsing system 10. In general, computer-based system 80 may include, among other things, a number of processing facilities, storage facilities, and work stations. As depicted, computer-based system 80 comprises a router/firewall 82, a load balancer 84, an Internet accessible network 86, an automated speech recognition (ASR)/text-to-speech (TTS) network 88, a telephony network 90, a database server 92, and a resource manager 94.

These computer-based system 80 may be deployed as a cluster of networked servers. Other clusters of similarly configured servers may be used to provide redundant processing resources for fault recovery. In one embodiment, each server may comprise a rack-mounted Intel Pentium processing system running Windows NT, Linux OS, UNIX, or any other suitable operating system.

For purposes of the present invention, the primary processing servers are included in Internet accessible network 86, automated speech recognition (ASR)/text-to-speech (TTS)

network 88, and telephony network 90. In particular, Internet accessible network 86 comprises one or more Internet access platform (IAP) servers. Each IAP servers implements the browser functionality that retrieves and parses conventional markup language documents supporting web pages. Each IAP servers builds the navigation trees 50 (which are the semantic representations of the web pages) and generates the navigation dialog with users. Telephony network 90 comprises one or more computer telephony interface (CTI) servers. Each CTI server connects the cluster to the telephone network which handles all call processing. ASR/TTS network 88 comprises one or more automatic speech recognition (ASR) servers and text-to-speech (TTS) servers. ASR and TTS servers are used to interface the text-based input/output of the IAP servers with the CTI servers. Each TTS server can also play digital audio data.

Load balancer 84 and resource manager 94 may cooperate to balance the computational load throughout computer-based system 10 and provide fault recovery. For example, when a CTI server receives an incoming call, resource manager 94 assigns resources (e.g., ASR server, TTS server, and/or IAP server) to handle the call. Resource manager 94 periodically monitors the status of each call and in the event of a server failure, new servers can be dynamically assigned to replace failed components. Load balancer 84 provides load balancing to maximize resource utilization, reducing hardware and operating costs.

Computer-based system 80 may have a modular architecture. An advantage of this modular architecture is flexibility. Any of

these core servers--i.e., IAP servers, CTI servers, ASR servers, and TTS servers--can be rapidly upgraded ensuring that voice browsing system 10 always incorporate the most up-to-date technologies.

5

#### Method For Browsing Content With Voice Commands

FIG. 8 is a flow diagram of an exemplary method 100 for browsing content with voice commands, according to an embodiment of the present invention. Method 100 may correspond to an aspect of operation of web browsing system 10.

Method 100 begins at step 102 where voice browsing system 10 receives at gateway module 30 a call from a user, for example, via a limited display device 18. In the call, the user may convey or issue a command or request. Such command or request can be in the form of voice or speech, and may pertain to particular content 15. This content 15 may be contained in a web page at a website or portal maintained by a content provider 12. Automatic speech recognition (ASR) component 36 of gateway module 30 operates on the voice/speech to recognize the user's command or request for content 15. Gateway module 30 forwards the request to browser module 32.

At step 104, in response to the request, voice browsing system 10 initiates a web browsing session for this interaction with the user. At step 106, browser module 32 loads or fetches a markup language document 58 supporting the web page that contains the desired content 15. This markup language document can be, for example, an HTML or an XML document. Browser module 32 may

also load or retrieve one or more style sheet documents 62 which are associated with the markup language document 58. At step 108, browser module 32 adds an identifier (e.g., a uniform resource locator (URL)) for the web page to a list maintained  
5 within voice browsing system 10. This is done so that voice browsing system 10 can keep track of each web page from which it has retrieved content; thus, at least some of the operations which voice browsing system 10 performs for any given web in response to an initial request do not need to be repeated in  
10 response to future requests relating to the same web page.

At step 110, navigation tree builder component 40 of browser module 32 builds a navigation tree 50 for the target web page. In one embodiment, to accomplish this, navigation tree builder component 40 may generate a document tree 60 from the  
15 conventional markup language document 58 and a style tree 64 from the style sheet document 62. The document tree 60 is then converted into the navigation tree 50, in part, using the style tree 64. The navigation tree 50 provides a semantic representation of the content contained in the target web page  
20 that is suitable for voice or audio commands. The navigation tree 50 comprises a plurality of nodes. Each such node may contain a portion of the content of the target web page or may provide prompts for directing the user to content. As such, navigation tree 50 enables a user to readily browse the content  
25 15 of the web page with voice commands and requests.

At step 112, navigation agent component 42 of browser module 32 begins browsing of the content by starting at a root node of

the navigation tree 50. The root node may comprise a number of different options from which a user can select, for example, to obtain content or to move to another node. To present these various options of the root node to the user, text-to-speech  
5 (TTS) component 38 of gateway module 30 may generate speech for the options, which is then delivered to the user via limited display device 18.

The user may then select one of the presented options, for example, by issuing a voice command which is recognized by  
10 automatic speech recognition component 36. At step 114, browsing module 32 moves to or "visits" the node of navigation tree 50 which is related to the user's selection.

At step 116, navigation agent component 42 determines whether the visited node is a routing node. A routing node is a  
15 node which may comprise a plurality of options from which the user may select in order to navigate through the navigation tree 50 in order to get to other nodes. If it is determined that the visited node is a routing node, then at step 118 browsing module 32 generates various prompts based upon the options available in  
20 the routing node. At step 120, text-to-speech component 38 plays the prompts to a user, for example, via limited display device 18. At step 122, gateway module 30 collects user input in response to the prompts played by text-to-speech component 38. This user input may specify or select among the various options  
25 offered by the prompts. At step 124, browser module 32 sets the current node to that node of navigation tree 50 which coincides

with the user's choice. Method 100 then moves to step 114, where browser module 32 visits the node of the user's choice.

On the other hand, if it is determined at step 116 that the current node is not a routing node, then at step 126 browser  
5 module 32 determines whether the current node is a form node. A form node is a node that relates to a respective form for collecting information or data. Such form may comprise a number of fields that can be filled out by a user. An exemplary form can be an order form which may be filled out in order to complete  
10 an electronic transaction via the website or portal associated with content provider 12.

If it is determined at step 126 that the current node is a form node, then method 100 moves to step 128 where browser module 32 and gateway module 30 cooperate to generate a dialog for  
15 filling out the respective form. The dialog may involve a series of questions that can be presented to a user by text-to-speech component 38. That is, in the dialog, text-to-speech component 30 may issue a number of prompts asking the user for input to fill in various fields of the form. At step 130, gateway module  
20 30 collects input from the user. One or more voice macros can be used to facilitate input collection. Voice macros map complex input to a simple voice command, increasing the convenience of data collection and improving the performance of a speech recognition task. For example, a voice macro can be created to  
25 map a user's credit card number to the phrase "my credit card." Then, when prompted at a form, the user may enter his or her credit card number simply by saying "my credit card".



At step 132, browser module 32 determines whether the form has been completed. If the form has not been completed, then method 100 returns to step 128, where the voice browsing system 10 continues to generate a dialog for filling out the form. In this case, prompts are played only for those fields that have not yet been filled out in the form. Steps 128 through 132 are repeated until the form is completed. When it is determined at step 132 that the form has been completed, then at step 134 voice browsing system 10 may submit the completed form to content provider 12 for further processing. The content provider 12 may then, for example, initiate completion of an electronic transaction, for example, by directing that a particular good be shipped to the user.

Referring again to step 126, if it is determined that the current node is not a form node, then at step 136 voice browsing system 10 determines whether the current node is a content node. A content node generally comprises information or content that can be presented to a user. If the current node is a content node, then at step 138 voice browsing system 10 plays the content to the user, for example, using text-to-speech component 38. Afterwards, method 100 returns to step 114, where another node is visited.

Otherwise, if it is determined at step 136 that the current node is not a content node, then at step 140 voice browsing system 10 determines whether the current node is a help node. A help node may comprise instructions to assist or guide the users during an interactive voice browsing session. If it is

determined that the current node is a help node, then at step 142 voice browsing system 10 plays the content of the help node in order to assist or guide the user. Afterwards, method 100 returns to step 114, where another node is visited.

5        On the other hand, if it is determined at step 140 that the current node is not a help node, then at step 144 voice browsing system 10 determines whether the current node is unknown to the system. A node may be unknown to the system if the command or request received from the user is indecipherable or unknown. If  
10    the current node is unknown, then voice browsing system 10 may deliver an appropriate message or prompt for notifying the user of such fact. At step 146 voice browsing system 10 computes the next page to be presented to a user. This can be done to inform the user that the current selection or request is not  
15    appropriate. After the next page has been computed or calculated, method 100 moves to step 106, where the conventional markup language document 58 supporting the computed next page is retrieved. Otherwise, if it is determined at step 144 that the current node is not an unknown node, then at step 148 voice  
20    browsing system 10 determines whether the current interactive session with the user should be ended. This can be done if, for example, a predetermined time has elapsed in which a user has not responded or, alternatively, if the user has actively taken action to end the session, such as, for example, hanging up. If  
25    it is determined that the current session should not be ended, then method 100 returns to step 114, where another node is

visited. Otherwise, if it is determined that the session should be ended, then method 100 ends.

Various steps in method 100 may be repeated throughout an interactive session with a user to generate navigation trees 50 and allow a user to obtain content and to move throughout the nodes of each navigation tree 50. Thus, a user is able to browse the content available at the web pages of a website or portal maintained by content provider 12 using voice commands or speech commands. This can be done with the existing infrastructure of conventional markup language documents of the website. Accordingly, content provider 12 is not required to set up and maintain a separate site in order to provide access and content to users.

#### 15 Method For Generating a Navigation Tree

FIG. 9 is a flow diagram of an exemplary method 200 for generating a navigation tree 50, according to an embodiment of the present invention. Method 200 may correspond to the operation of navigation tree builder component 40 of browser module 32.

Method 200 begins at step 202 where navigation tree builder component 40 receives a conventional markup language document 58 from a content provider 12. The conventional markup language document, which may support a respective web page, may comprise content 15 and formatting for the same. At step 204, markup language parser 52 parses the elements of the received markup language document 58. For example, content 15 in the markup

language document 58 may be separated from formatting tags. At step 206, markup language parser 52 generates a document tree 60 using the parsed elements of the conventional markup language document 58.

5       At step 208, navigation tree builder component 40 receives a style sheet document 62 from the same content provider 12. This style sheet document 62 may be associated with the received conventional markup language document 58. The style sheet document 62 provides metadata, such as declarative statements  
10 (rules) and procedural statements. At step 210, style sheet parser 54 parses the style sheet document 62 to generate a style tree 64.

Tree converter 56 receives the document tree 60 and the style tree 64 from markup language parser 52 and style sheet  
15 parser 54, respectively. At step 212, tree converter 56 generates a navigation tree 50 using the document tree 60 and the style tree 64. In one embodiment, among other things, tree converter 56 may apply style sheet rules and heuristic rules to the document tree 60, and map elements of the document tree 60  
20 into nodes of the navigation tree 50. Afterwards, method 200 ends.

#### Method For Applying Style Sheet Rules To a Document Tree

FIG. 10 is a flow diagram of an exemplary method 300 for  
25 applying style sheet rules to a document tree 60, according to an embodiment of the present invention. Method 300 may correspond to the operation of style sheet engine 68 in tree converter 56 of

voice browsing system 10. In general, style sheet engine 68 selects various nodes of a document tree 60 and applies style sheet rules to these nodes as part of the process of converting the document tree 60 into a navigation tree 50.

5       Method 300 begins at step 302, where selector module 74 of style sheet engine 68 selects various nodes of a document tree 60 for clipping. As used herein, clipping may comprise saving the various selected nodes so that these nodes will remain or stay intact during the transition from document tree 60 into  
10   navigation tree 50. Nodes are clipped if they are sufficiently important. At step 304, rule applicator module 76 clips the selected nodes.

At step 306, selector module 74 selects various nodes of the document tree 60 for pruning. As used herein, pruning may  
15   comprise eliminating or removing certain nodes from the document tree 60. For example, nodes are desirably pruned if they have content (e.g., image or animation files) that is not suitable for audio presentation. At step 308, rule applicator module 76 prunes the selected nodes.

20       At step 310, selector module 74 of style sheet engine 68 selects certain nodes of the document tree for filtering. As used herein, filtering may comprise adding data or information to the document tree 60 during the conversion into a navigation tree 50. This can be done, for example, to add information for a  
25   prompt or label at a node. At step 312, rule applicator module 76 filters the selected nodes.

At step 314, selector module 74 selects certain nodes of document tree 60 for conversion. For example, a node in a document tree having content arranged in a table format can be converted into a routing node for the navigation tree. At step 5 316, rule applicator module 76 converts the selected nodes. Afterwards, method 300 ends.

#### Method For Applying Heuristic Rules To a Document Tree

FIG. 11 is a flow diagram of an exemplary method 400 for 10 applying heuristic rules to a document tree 60, according to an embodiment of the present invention. In one embodiment, method 400 may correspond to the operation of heuristic engine 70 in tree converter 56 of voice browsing system 10. These heuristic rules can be learned by heuristic engine 70 during the operation 15 of voice browsing system 10. Each of the heuristic rules can be applied separately to various nodes of the document tree 60. Application of heuristic rules can be done on a node-by-node basis during the transformation of a document tree 60 into a navigation tree 50.

20 Method 400 begins at step 402, where heuristic engine 70 selects a node of document tree 60. At step 404, heuristic engine 70 may convert page and line breaks in the content contained at such node into white space. This is done to eliminate unnecessary formatting and yet not concatenate content 25 (e.g., text). At step 406, heuristic engine 70 exploits image alternative tags within the content of a web page. These image alternative tags generally point to content which is provided as

an alternative to images in a web page. This content can be in the form of text which is read or spoken to a user with a hearing impairment (e.g., deaf). Since this alternative content is appropriate for delivery by speech or audio, heuristic engine 70  
5 exploits the image alternative tags.

At step 408, if the node is decorative, heuristic engine 70 deletes such node from the document tree 60. In one embodiment, nodes may be considered to be decorative if they do not provide any useful function in a navigation tree 50. For example, a  
10 content node consisting of only an image file may be considered to be decorative since the image cannot be presented to a user in the form of speech or audio.

At step 410, heuristic engine 70 merges together content and associated links at the node in order to provide a continuous  
15 flow of data to a user. Otherwise, the internal links would act as disruptive breaks during the delivery of content to users. At step 412, heuristic engine 70 builds outlines of headlines and ordered lists in the document tree.

After all applicable heuristic rules have been applied to  
20 the current node, then at step 414 heuristic engine 70 determines whether there are any other nodes in the document tree 60 which should be processed. If there are additional nodes, then method 400 returns to step 402, where the next node is selected. Steps 402 through 414 are repeated until the heuristic rules are  
25 applied to all nodes of the document tree 60. When it is determined at step 414 that there are no other nodes in the document tree, method 400 ends.

Method For Mapping a Document Tree Into a Navigation Tree

FIG. 12 is a flow diagram of an exemplary method 500 for mapping a document tree 60 into a navigation tree 50, according to an embodiment of the present invention. Method 500 may correspond to the operation of mapping engine 72 in tree converter 56 of navigation tree builder component 40. Method 500 may be performed on a node-by-node basis during the transformation of a document tree 60 into a navigation tree 50.

Method 500 begins at step 502, where mapping engine 72 selects a node of the document tree 60. At step 504, mapping engine 72 determines whether the selected node contains content. If the selected node contains content, then at step 506 mapping engine 72 creates a content node in the navigation tree 50. A content node of the navigation tree 50 comprises content that can be presented or played to a user, for example, in the form of speech or audio, during navigation of the navigation tree 50. Afterwards, method 500 returns to step 502, where the next node in the document tree is selected.

Otherwise, if it is determined at step 504 that the current node is not a content node, then at step 508 mapping engine 72 determines whether the selected node contains an ordered list, an unordered list, or a table row. If the currently selected node comprises an ordered list, an unordered list, or a TR, then at step 510 mapping engine 72 creates a suitable routing node for the navigation tree 50. Such routing node may comprise a plurality options which can be selected in the alternative to



move to another node in the navigation tree 50. Afterwards, method 500 returns to step 502, where the next node is selected.

On the other hand, if it is determined at step 508 that the currently selected node does not contain any of an ordered list, an unordered list, or a TR, then at step 512 mapping engine 72 determines whether the currently selected node of the document tree is a node for a table. If it is determined at step 512 that the node is a table node, then at step 514 mapping engine 72 creates a suitable table node for the navigation tree 50. A table node in the navigation tree 50 is used to hold an array of information. A table node in navigation tree 50 can be a routing node. Afterwards, method 500 returns to step 502, where the next node is selected.

Alternatively, if it is determined at step 512 that the currently selected node is not a table node, then at step 516 mapping engine 72 determines whether the node of the document tree 60 contains a form. Such form may have a number of fields which can be filled out in order to collect information from a user. If it is determined that the current node of the document tree 40 contains a form, then at step 518 mapping engine 72 creates an appropriate form node for the navigation tree 50. A form node may comprise a plurality prompts which assist a user in filling out fields. Afterwards, method 500 returns to step 502, where the next node is selected.

Otherwise, if it is determined at step 516 that the current node does not contain a form, then at step 520 mapping engine 72 determines whether there are form elements at the node. Form

elements can be used to collect input from a user. The information is then sent to be processed by a Web server. If there are form elements at the node, then at step 522 mapping engine 72 maps a form handling node to the form elements. Form  
5 handling nodes are provided in navigation tree 50 to collect input. This can be done either with direct input or with voice macros. Afterwards, method 500 returns to step 502 where another node is selected.

On the other hand, if it is determined at step 520 that the  
10 current node of the document tree 60 does not contain form elements, then at step 524 mapping engine 72 determines whether there are any more nodes in the document tree 60. If there are other nodes, then method 500 returns to step 502, where the next node is selected. Steps 502 through 524 are repeated until  
15 mapping engine 72 has processed all nodes of the document tree 60, for example, to map suitable nodes into navigation tree 50. Thus, when it is determined at step 524 that there are no other nodes in the document tree, method 500 ends.

20 Although particular embodiments of the present invention have been shown and described, it will be obvious to those skilled in the art that changes and modifications may be made without departing from the present invention in its broader aspects, and therefore, the appended claims are to encompass  
25 within their scope all such changes and modifications that fall within the true scope of the present invention.

Appendix AClasses/Types of Nodes

There are two broad classes of nodes found in a navigation  
5 tree: routing nodes and content nodes. Routing nodes can be of  
different types, including, for example, general routing nodes,  
group nodes, input nodes, array nodes, and form nodes. Content  
nodes can also be of different types, including, for example,  
text and element. The allowable children type for each node can  
10 be as follows:

General Routing Node <ROUTE> Group Node, Routing Node  
Group Node <GROUP>: Content Node, Group Node  
Input Node <INPUT>: Content  
15 Array Node <ARRAY>: Group Node  
Form Node <FORM>: Input Node  
Text Node <TEXT>  
Element Node <ELEM>

20 Each of the routing node types can be "visited" by a tree  
traversal operation, which can be either step navigation or rapid  
access navigation. General routing nodes (<ROUTE>) permit  
stepping to their children. Group nodes (<GROUP>) do not permit  
stepping to their children.

25 Content nodes are the container objects for text and markup  
elements. Content nodes are not routing nodes and hence are not  
reachable other than through a routing node. Note that all  
content nodes should have a group node for a parent. A group  
node can retrieve data contained in the children content nodes.

Element nodes correspond to various generic tags including anchor, formatting, and unknown tags. Element nodes can be implemented either by retaining an original SGML/XML tag or setting a tag attribute of the <ELEM> markup tag could contain to  
 5 the SGML/XML tag.

### Data fields

Every node has a basic set of attributes. These attributes can be used to generate interactive dialogs (e.g., voice commands  
 10 and speech prompts) with the user.

```

    // Attributes used by style sheet
String class;           // class attribute
15 String id;            // id attribute
String style;           // style attributes

    // Properties best defined in a style sheet
20 String element;       // tag element of node
String node-type;       // node type (e.g., Routing)
  
```

The "element" attribute stores the name of an SGML/XML element  
 25 tag before conversion into the navigation tree. The "class" and "id" attributes are labels that can be used to reference the node. The "style" attribute specifies text to be used by the style sheet parser.

### 30 Group Node

A group node is a container for text, links, and other markup elements such as scripts or audio objects. A contiguous block of unmarked text, structured text markup, links, and text

formatting markup are parsed into a set of content nodes. The group node is a parent that organizes these content nodes into a single presentational unit.

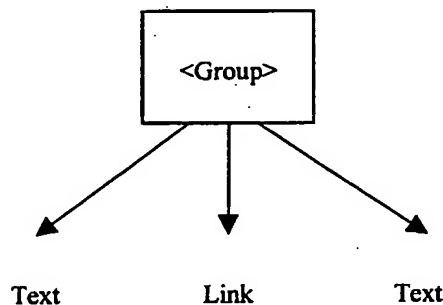
For example, the following HTML line:

5 Go to <A HREF = "http://www.vocalpoint.com"> Vocal Point </A>.

could be parsed into the form shown below:

10 <GROUP>  
Go to <A HREF = http://www.vocalpoint.com> Vocal Point </A> .  
</GROUP>

This particular group node specifies that the three children  
15 nodes "Go to", anchor link "Vocal Point", and "." should be  
presented as a single unit, not separately.



A group node does not allow its children to be visited by a tree traversal operation. Content nodes can only have group  
20 nodes for parents. Consequently, content nodes are not directly  
reachable, but rather can only be accessed from the parent group  
node.

A group node can sometimes be the child of another content group. In this case, the child group node is also unreachable by

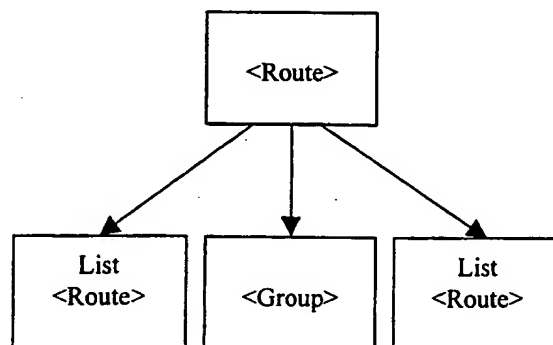
tree traversal operations. A special class of group node called an array node must be used to access data in nested group nodes.

### Input Node

- 5        An input node is similar to a group node except for two differences. First, an input node can retrieve and store input from the user. Second, an input node can only be a child of a form node.

### 10    General routing Node

- A general routing node is the basic building block for constructing hierarchical menus. General routing nodes serve as way points in the navigation tree to help guide users to content. The children of general routing nodes are other general routing nodes or group nodes. When visited, a general routing node will supply prompt cues describing its children. An exemplary structure for a general routing node and its children is as follows:



### Array Node

An array node is used to build a multi-dimensional array representation of content. The HTML <TABLE> tag directly maps to an array node. To build up an array node from a document tree,  
5 information is extracted from the children element nodes.

### Form Node

A form node is a parent of an input node. Form nodes collect input information from the user and execute the  
10 appropriate script to process the forms. Form nodes also control review and editing of information entered into the form. The HTML <FORM> tag directly maps to a form node.

### A Brief Introduction to HML

15 Hierarchical markup language (HML) is designed to provide a file representation of the navigation tree. HML uses the specification for XML. Content providers may create content files using HML or translation servers can generate HML files from HTML/XML and XCSS documents. HML documents provide  
20 efficient representations of navigation trees, thus reducing the computation time needed to parse HTML/XML and XCSS.

### Syntax

HML elements use the "hml" namespace. A list of these  
25 elements is provided below:

<hml:root>	Root of the navigation tree
<hml:route>	Routing node
<hml:group>	Group node
<hml:array>	Array node
<hml:input>	Input node
<hml:form>	Form node

### Abbreviated Document Type Definition

XML syntax is described using a document type definition (DTD). An abbreviated, partially complete, DTD for HML follows.

5

```
<!--===== Generic Attributes =====>
```

```
<!ENTITY % coreattrs
  "id          ID          # -- document-wide unique id
   class       CDATA       # -- space sep. list of classes
   style       %StyleSheet # -- associated style info"
>
```

10

```
<!ENTITY % navattrs
  keys          CDATA       # -- space sep. list of keys
  descriptor    CDATA       # -- short description of node
  prompt        CDATA       # -- prompt
  greeting      CDATA       # -- greeting
>
```

15

20

```
<!ENTITY % attrs "coreattrs; navattrs;" >
```

```
<!--===== Text Markup =====>
```

25

```
<!ENTITY % special "A | OBJECT | SCRIPT" >
```

```
<!ENTITY % inline "%PCDATA | %special;" >
```

30

```
<!--===== Content Group =====>
```

```
<!ELEMENT HML:GROUP - - (%inline;)* (GROUP)* -- content group -->
```

```
<!ATTLIST
  %attrs;
>
```

35

```
<!--===== Routing Node =====>
```



```
5      <!ELEMENT HML:ROUTE - - (%inline;)* (GROUP)* (ROUTE)* -- route -->
      <!ATTLIST
        %attrs;
      >

10    <!--===== HTML Elements =====>

    <!ELEMENT A - - -- anchor -->
    <!ELEMENT OBJECT - - -- object -->
    <!ELEMENT SCRIPT - - -- script -->
```

WHAT IS CLAIMED IS:

1. A computer system for allowing a user of a limited display device to browse content available from a data network, the system comprising:

5 an interface operable to receive a request for the content from the user via the limited display device; and

a processor coupled to the interface, the processor operable to retrieve a conventional markup language document containing the content from the data network, the processor operable to  
10 convert the conventional markup language document into a navigation tree which provides a semantic, hierarchical structure for the content.

2. The computer system of Claim 1 wherein the limited  
15 display device comprises a wireless telephone, a smart telephone, or a wireless personal digital assistant (PDA).

3. The computer system of Claim 2 wherein the interface comprises a computer telephony interface (CTI)/personal digital  
20 assistant (PDA) component operable to support communication with the limited display device.

4. The computer system of Claim 1 wherein the interface comprises an automated speech recognition (ASR) component  
25 operable to recognize speech input from a user.

5. The computer system of Claim 1 wherein the interface comprises a text-to-speech component operable to output speech for the content.

5 6. The computer system of Claim 1 wherein the conventional markup language document comprises a Hypertext Markup Language (HTML) document.

7. The computer system of Claim 1 wherein the conventional  
10 markup language document comprises an eXtensible Markup Language (XML) document.

8. The computer system of Claim 1 wherein the processor is operable to retrieve a style sheet document from the data  
15 network, the style sheet document associated with the conventional markup language document and containing metadata which can be applied to the conventional markup language document.

20 9. The computer system of Claim 8 wherein the style sheet document comprises an extended Cascading Style Sheet (xCSS) document.

10. The computer system of Claim 8 wherein the processor is  
25 operable to generate a document tree from the conventional markup language document and to generate a style tree from the style sheet document.

11. The computer system of Claim 1 wherein the navigation tree comprises:

a plurality of content nodes comprising the content; and

5 at least one routing node comprising a plurality of options for moving between the nodes.

12. A method performed on a computer for allowing a user of a limited display device to browse content available from a data network, the method comprising:

receiving a request for the content from the user via the limited display device;

retrieving a conventional markup language document containing the content from the data network; and

15 converting the conventional markup language document into a navigation tree which provides a semantic, hierarchical structure for the content.

13. The method of Claim 12 wherein the request is in the form of speech, the method further comprising recognizing the speech.

14. The method of Claim 12 wherein the conventional markup language document comprises a Hypertext Markup Language (HTML) document.

15. The method of Claim 12 wherein the conventional markup language document comprises an eXtensible Markup Language (XML) document.

5 16. The method of Claim 12 further comprising retrieving a style sheet document from the data network, the style sheet document associated with the conventional markup language document and containing metadata which can be applied to the conventional markup language document.

10 17. The method of Claim 16 wherein the style sheet document comprises an extended Cascading Style Sheet (xCSS) document.

15 18. The method of Claim 16 wherein converting comprises:  
generating a document tree from the conventional markup language document; and  
generating a style tree from the style sheet document.

20 19. The method of Claim 12 wherein converting comprises:  
generating a document tree from the conventional markup language document; and  
applying a plurality of style sheet rules to the document tree.

25 20. The method of Claim 12 wherein the navigation tree comprises:  
a plurality of content nodes comprising the content; and

at least one routing node comprising a plurality of options for moving between the nodes.

21. The method of Claim 20 further comprising navigating  
5 between the content nodes of the navigation tree using the routing node.

22. A computer system for allowing a user of a limited display device to browse content available from a data network,  
10 the system comprising:

a markup language parser operable to receive a conventional markup language document in response to a request for the content from the user via the limited display device, the markup language parser operable to generate a document tree from the conventional  
15 markup language document;

a style sheet parser operable to receive a style sheet document in response to the request, the style sheet parser operable to generate a style tree from the style sheet document, the style tree comprising a plurality of style sheet rules; and

20 a tree converter in communication with the markup language parser and the style sheet parser, the tree converter operable to convert the document tree into a navigation tree using the style sheet tree rules, the navigation tree providing a semantic, hierarchical structure for the content.

25

23. The computer system of Claim 22 wherein the tree converter comprises a style sheet engine operable to apply the style sheet rules to the document tree.

5        24. The computer system of Claim 22 wherein the tree converter comprises a heuristic engine operable to apply a plurality of heuristic rules to the document tree.

10       25. The computer system of Claim 22 wherein the conventional markup language document comprises a Hypertext Markup Language (HTML) document.

15       26. The computer system of Claim 22 wherein the conventional markup language document comprises an eXtensible Markup Language (XML) document.

20       27. The computer system of Claim 22 wherein the style sheet document comprises an extended Cascading Style Sheet (xCSS) document.

28. A method performed on a computer for allowing a user of a limited display device to browse content available from a data network, the method comprising:

25       receiving a conventional markup language document and a style sheet document in response to a request for the content from the user via the limited display device;

generating a document tree from the conventional markup language document;

generating a style tree from the style sheet document, the style tree comprising a plurality of style sheet rules; and

5 converting the document tree into a navigation tree using the style sheet tree rules, the navigation tree providing a semantic, hierarchical structure for the content.

29. The method of Claim 28 wherein the conventional markup language document comprises a Hypertext Markup Language (HTML) document.

30. The method of Claim 28 wherein the conventional markup language document comprises an eXtensible Markup Language (XML) document.

31. The method of Claim 28 wherein the style sheet document comprises an extended Cascading Style Sheet (xCSS) document.

20 32. The method of Claim 28 wherein converting comprises a applying the style sheet rules to the document tree.

33. The method of Claim 28 wherein converting comprises applying a plurality of heuristic rules to the document tree.

25



34. A computer system for allowing a user of a limited display device to browse content available from a data network, the system comprising:

a gateway module operable to receive a spoken request for the content from the user via the limited display device, the gateway module operable to recognize the spoken request; and

a browser module in communication with the gateway module, the browser module operable to retrieve a conventional markup language document and a style sheet document from the data network in response to the spoken request, the conventional markup language document containing the content, the style sheet document containing metadata, the browser module operable to generate a navigation tree using the conventional markup language document and the style sheet document, the navigation tree providing a semantic, hierarchical structure for the content;

wherein the gateway module and the browser module are operable to cooperate to enable the user to browse the content using the navigation tree and to output speech conveying the content to the user via the limited display device.

35. The computer system of Claim 34 wherein the gateway module is operable to convert the spoken request into a text format.

36. A method performed on a computer for allowing a user of a limited display device to browse content available from a data network, the method comprising:

receiving a spoken request for the content from the user via the limited display device;

recognizing the spoken request;

retrieving a conventional markup language document and a  
5 style sheet document from the data network in response to the spoken request, the conventional markup language document containing the content, the style sheet document containing metadata;

generating a navigation tree using the conventional markup  
10 language document and the style sheet document, the navigation tree providing a semantic, hierarchical structure for the content;

enabling the user to browse the content using the navigation tree; and

15 outputting speech conveying the content to the user via the limited display device.

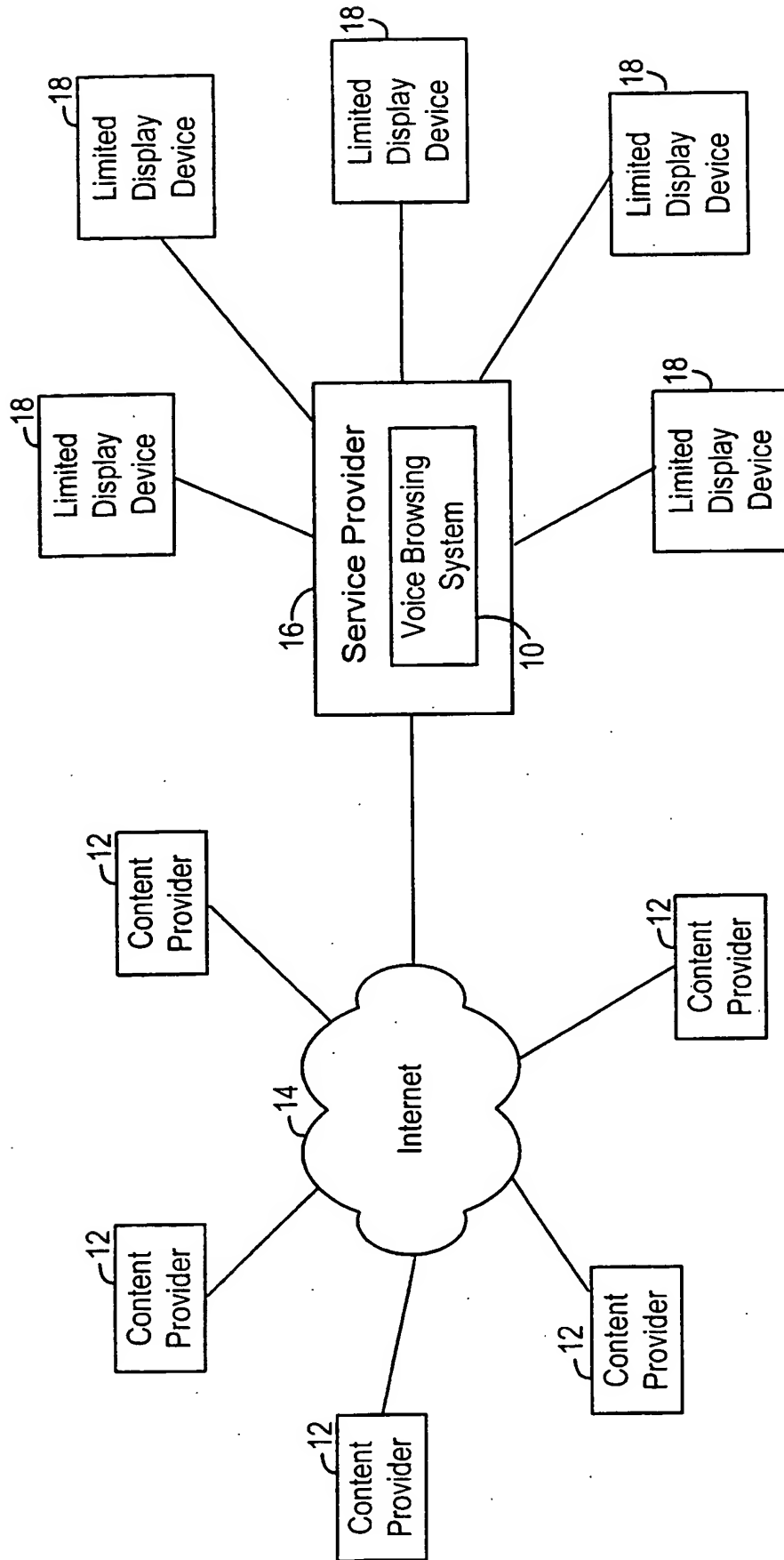


Fig. 1A

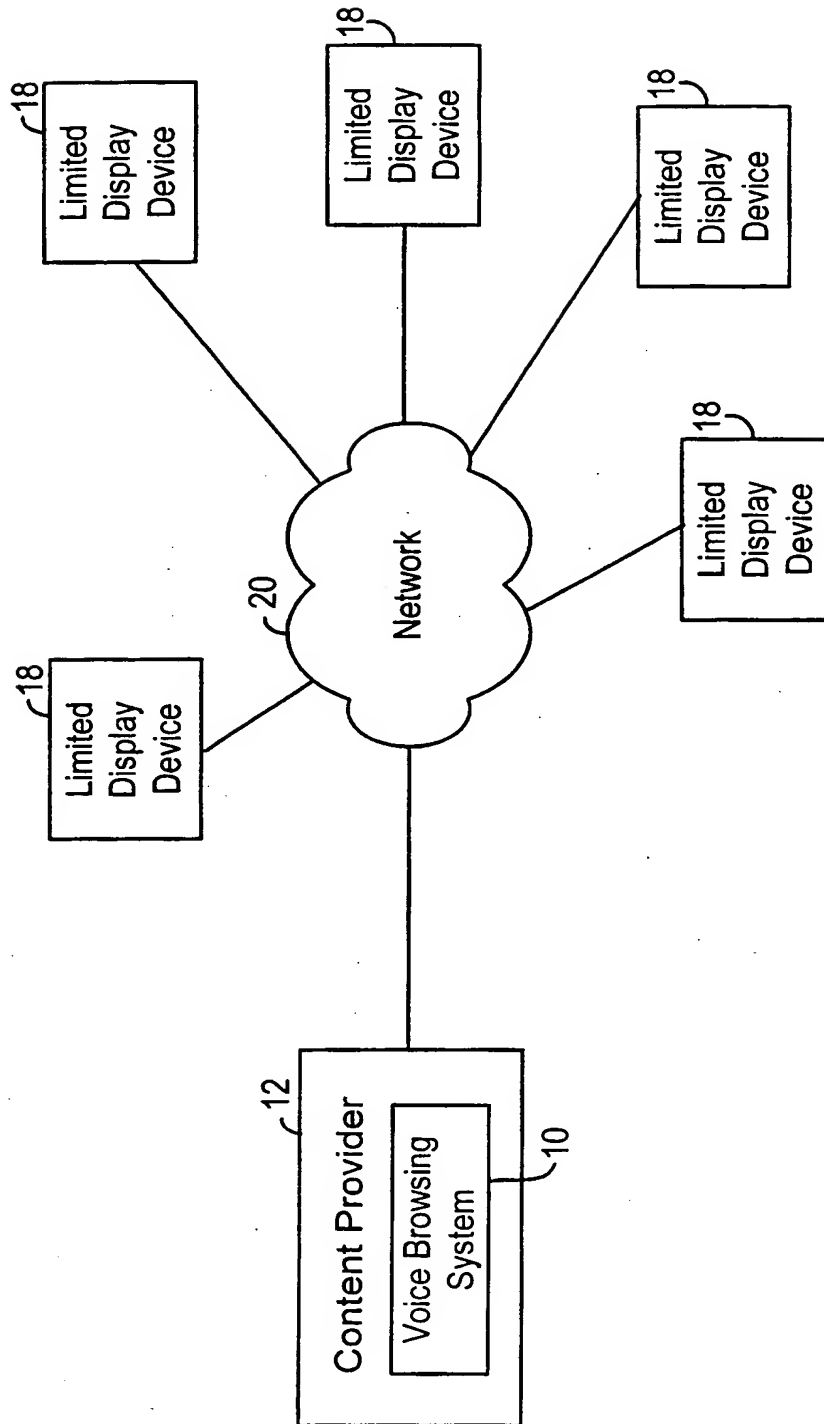


Fig. 1B

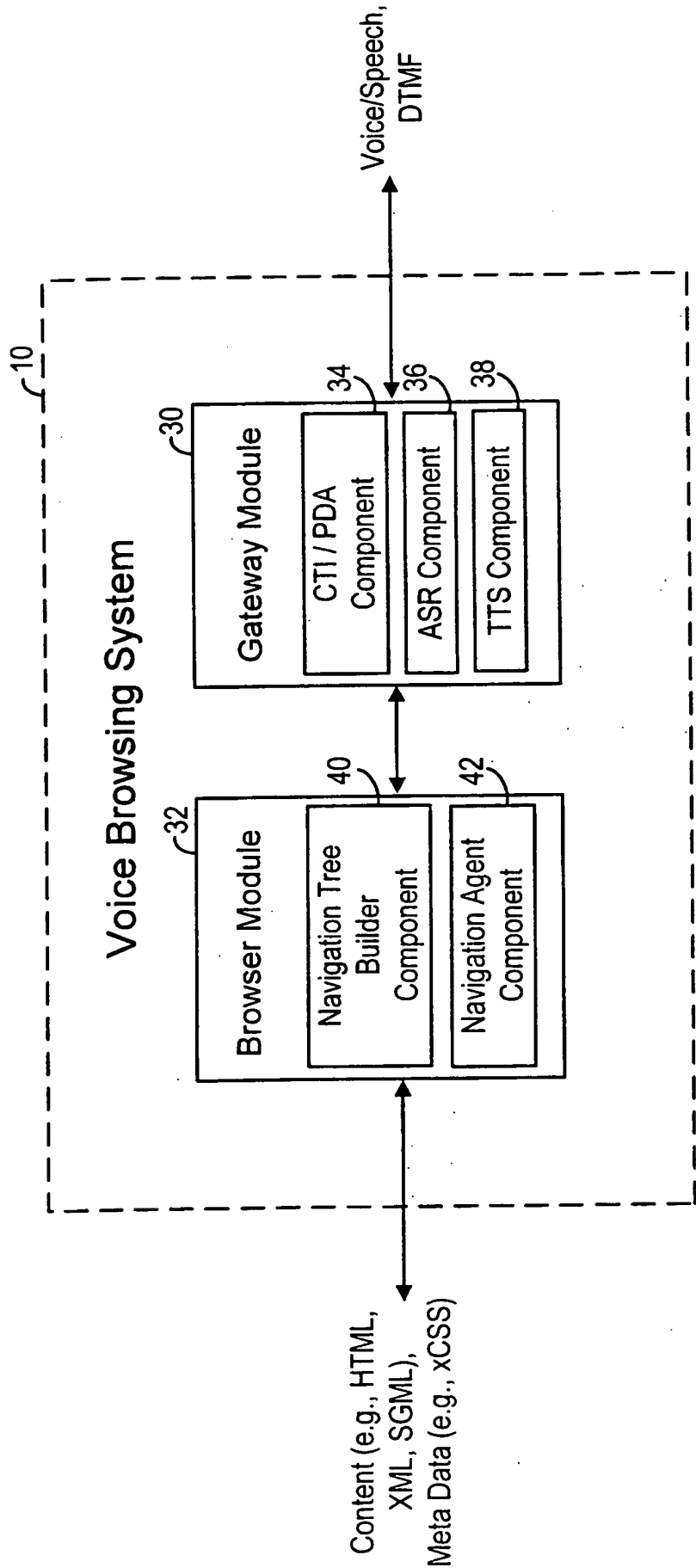


Fig. 2

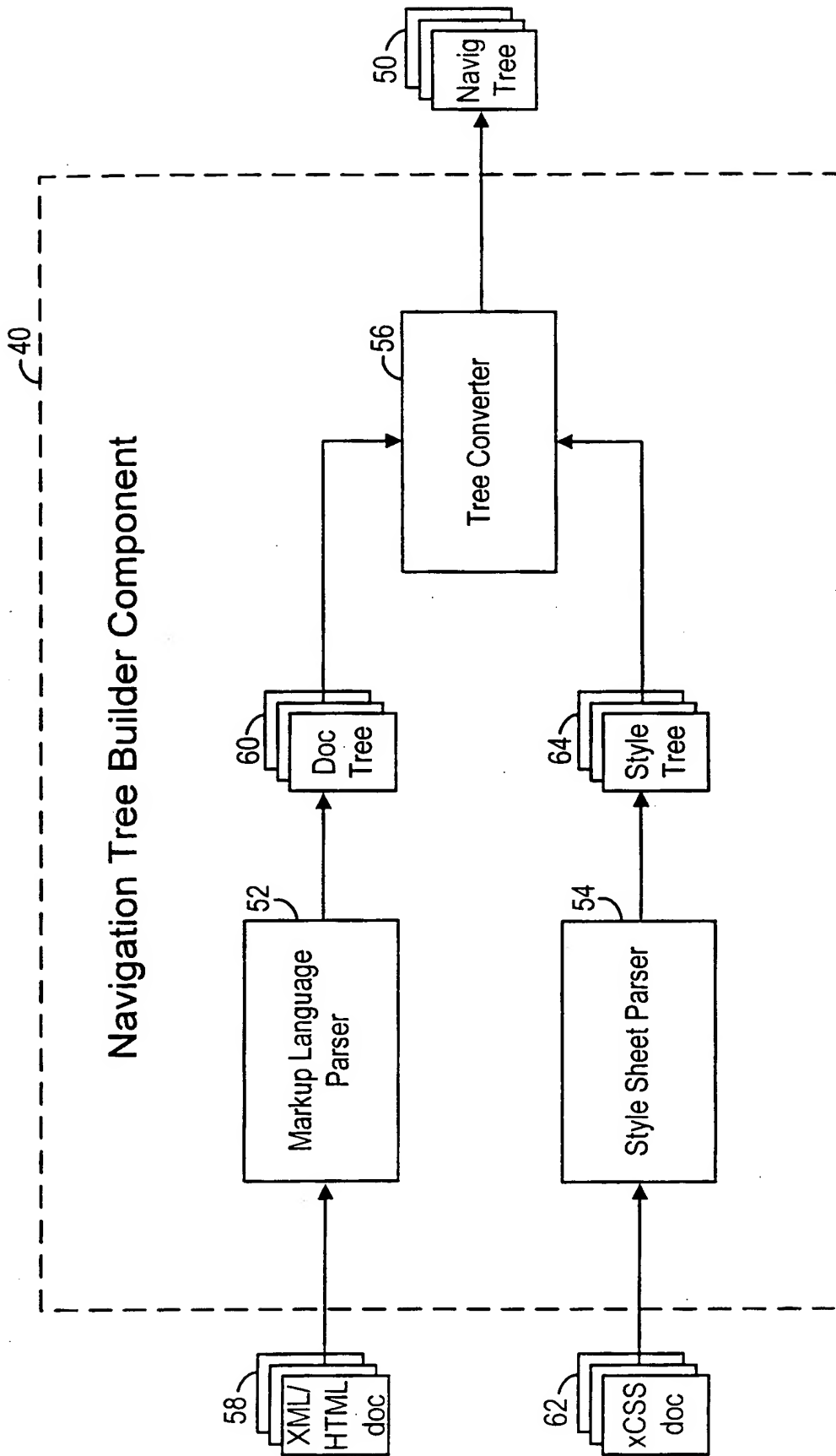


Fig. 3

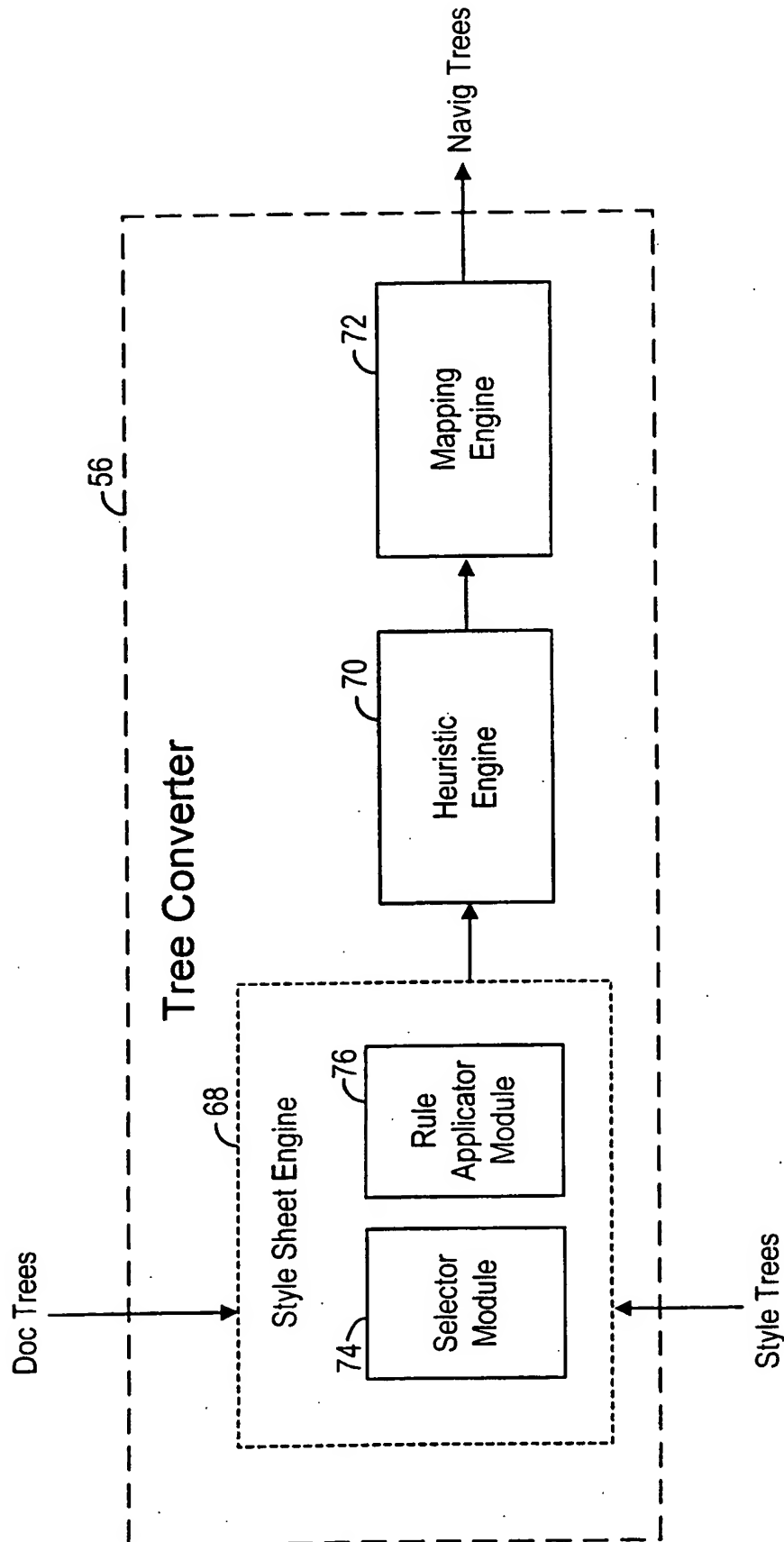


Fig. 4

## Document Tree

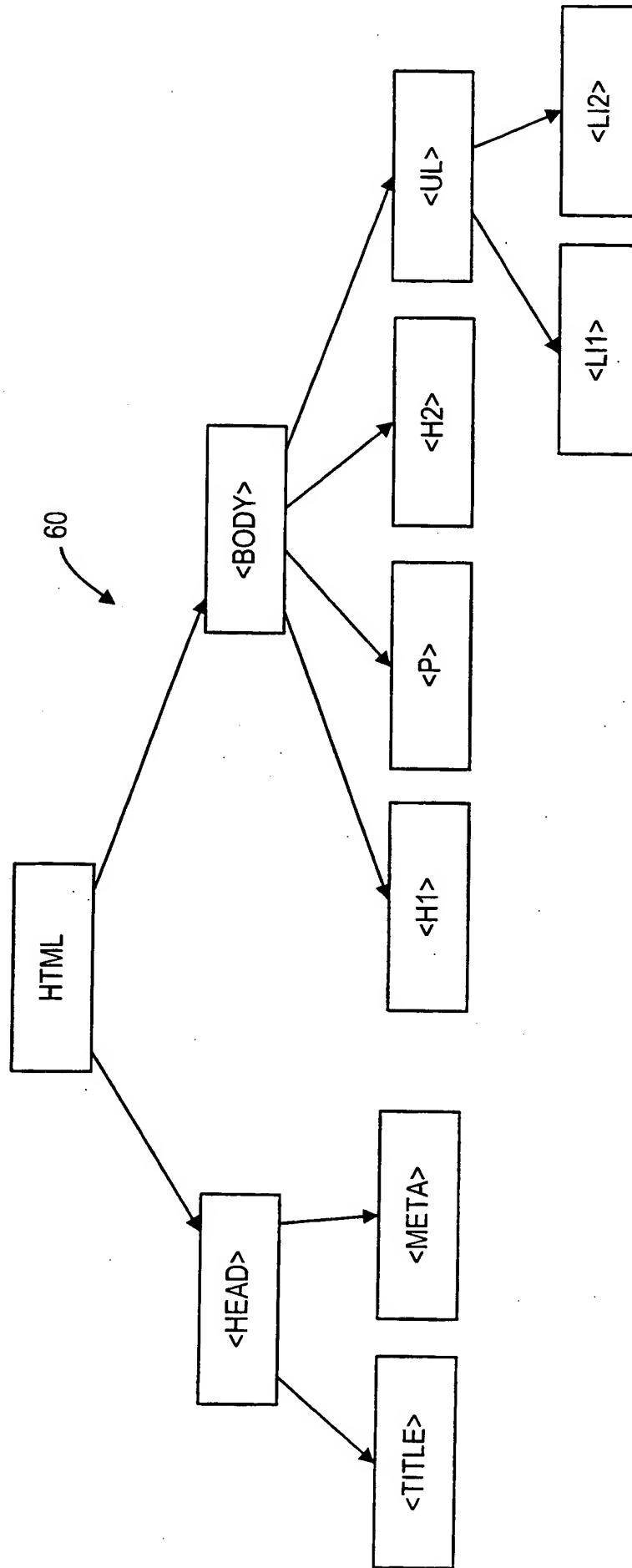


Fig. 5



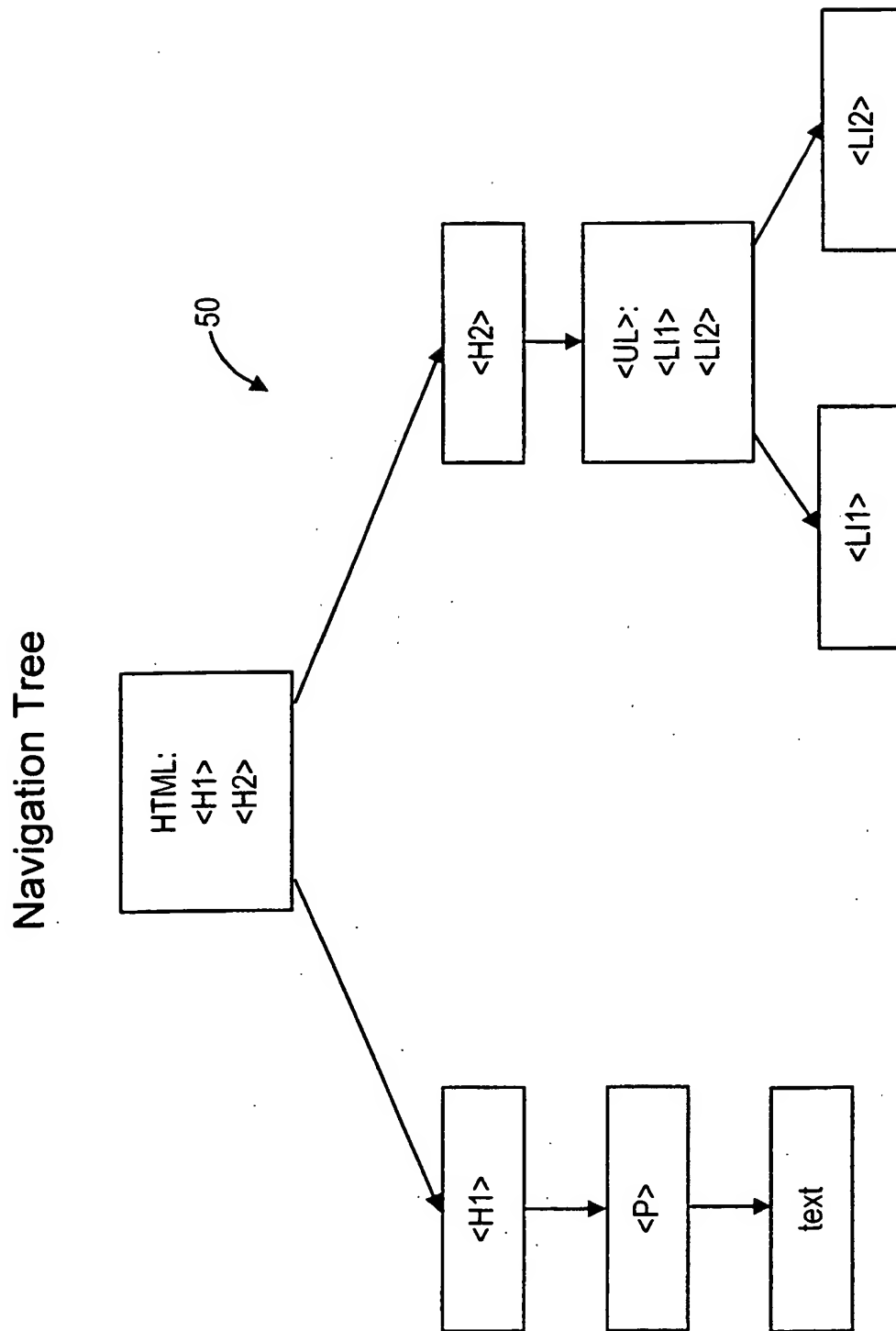


Fig. 6

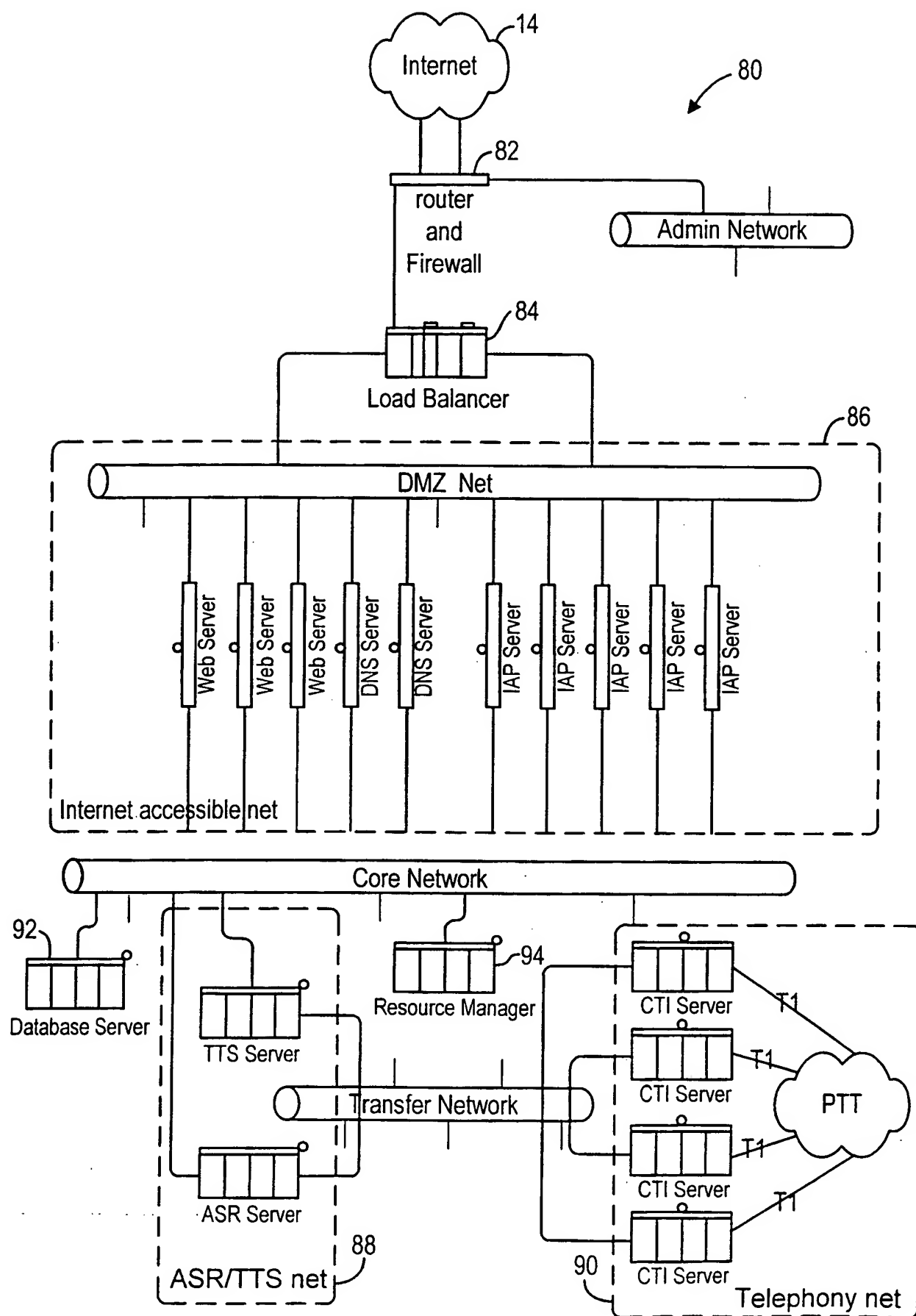


Fig. 7

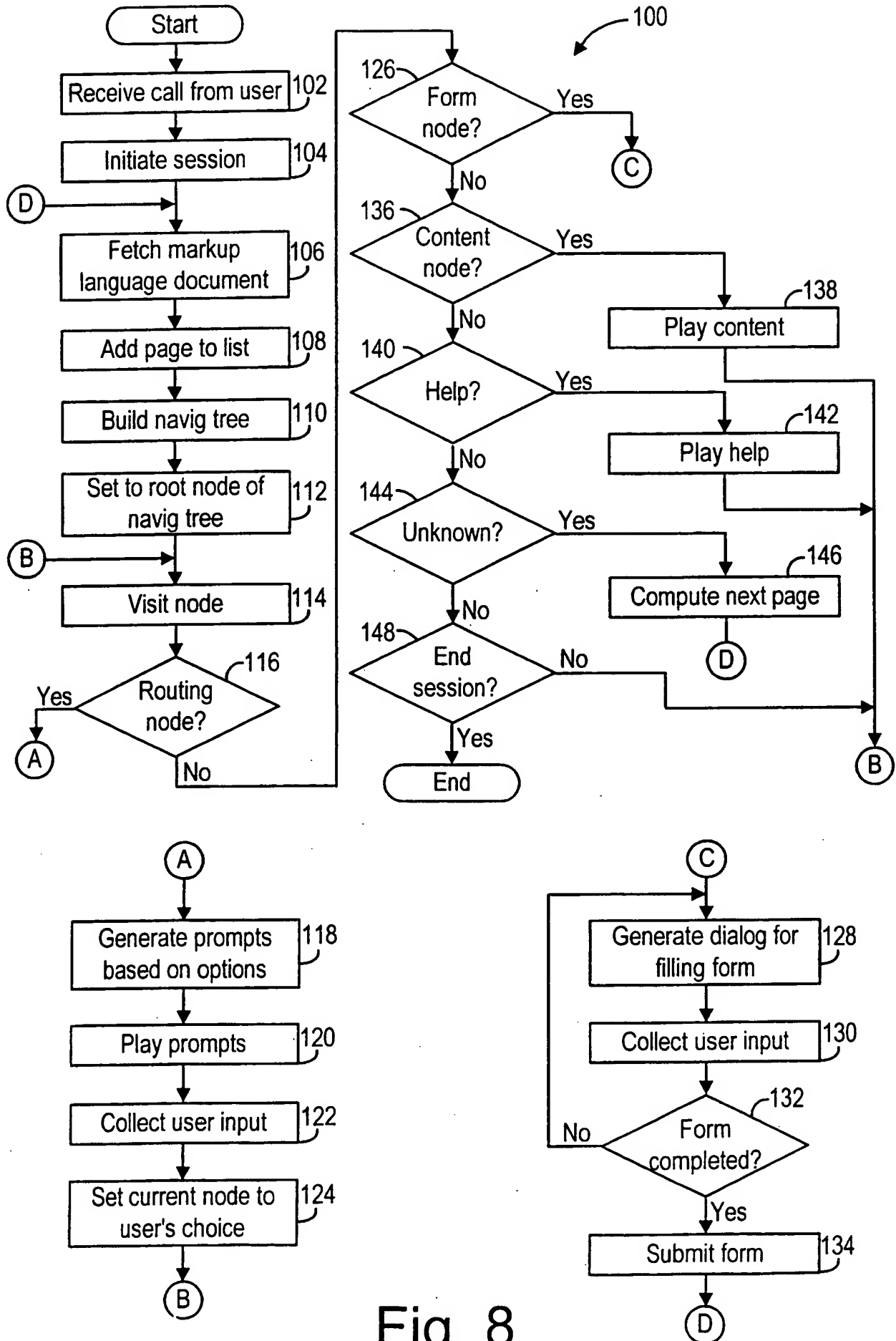


Fig. 8

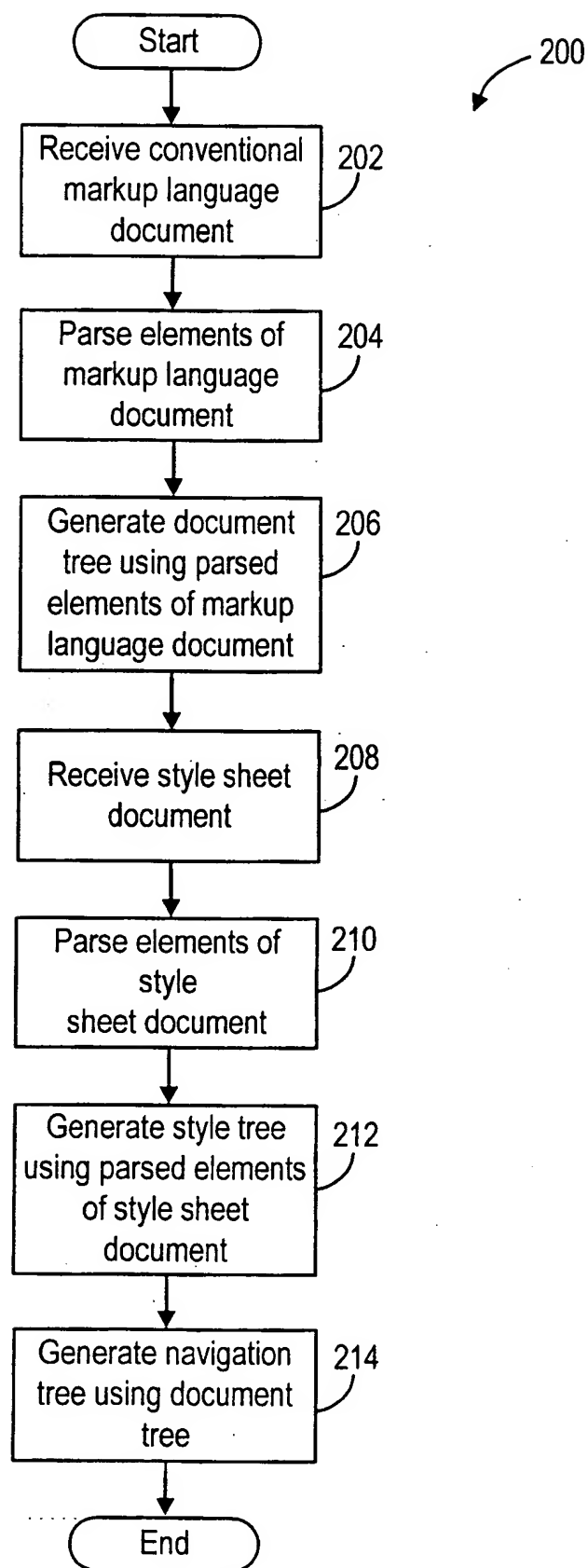


Fig. 9

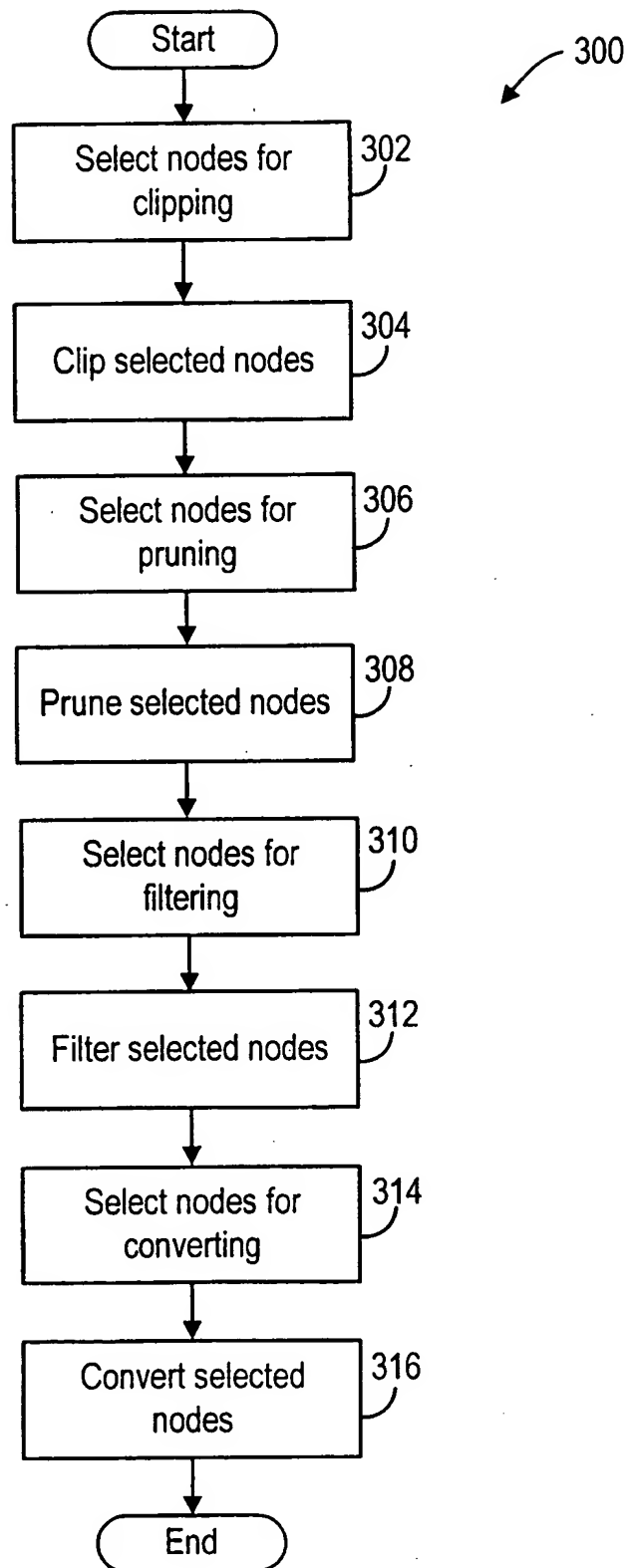


Fig. 10

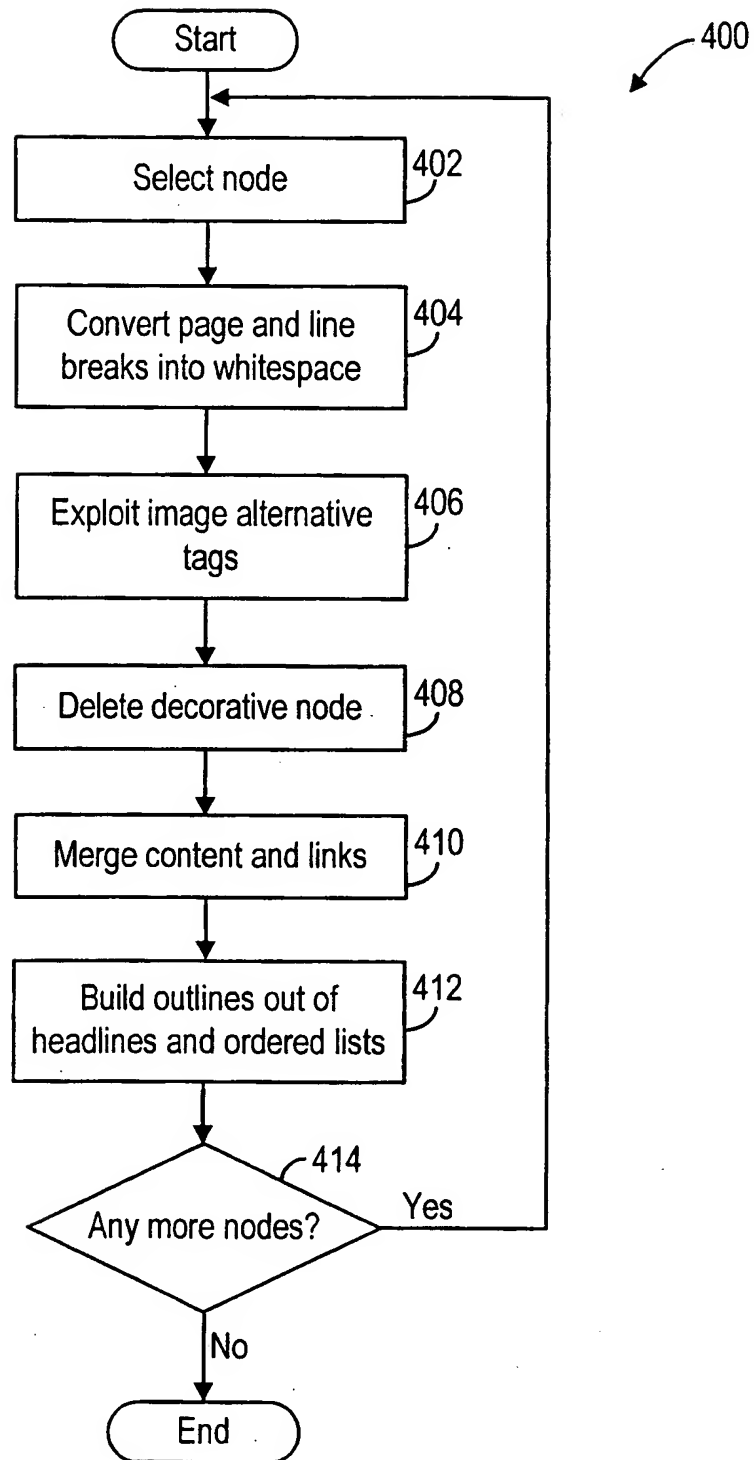


Fig. 11

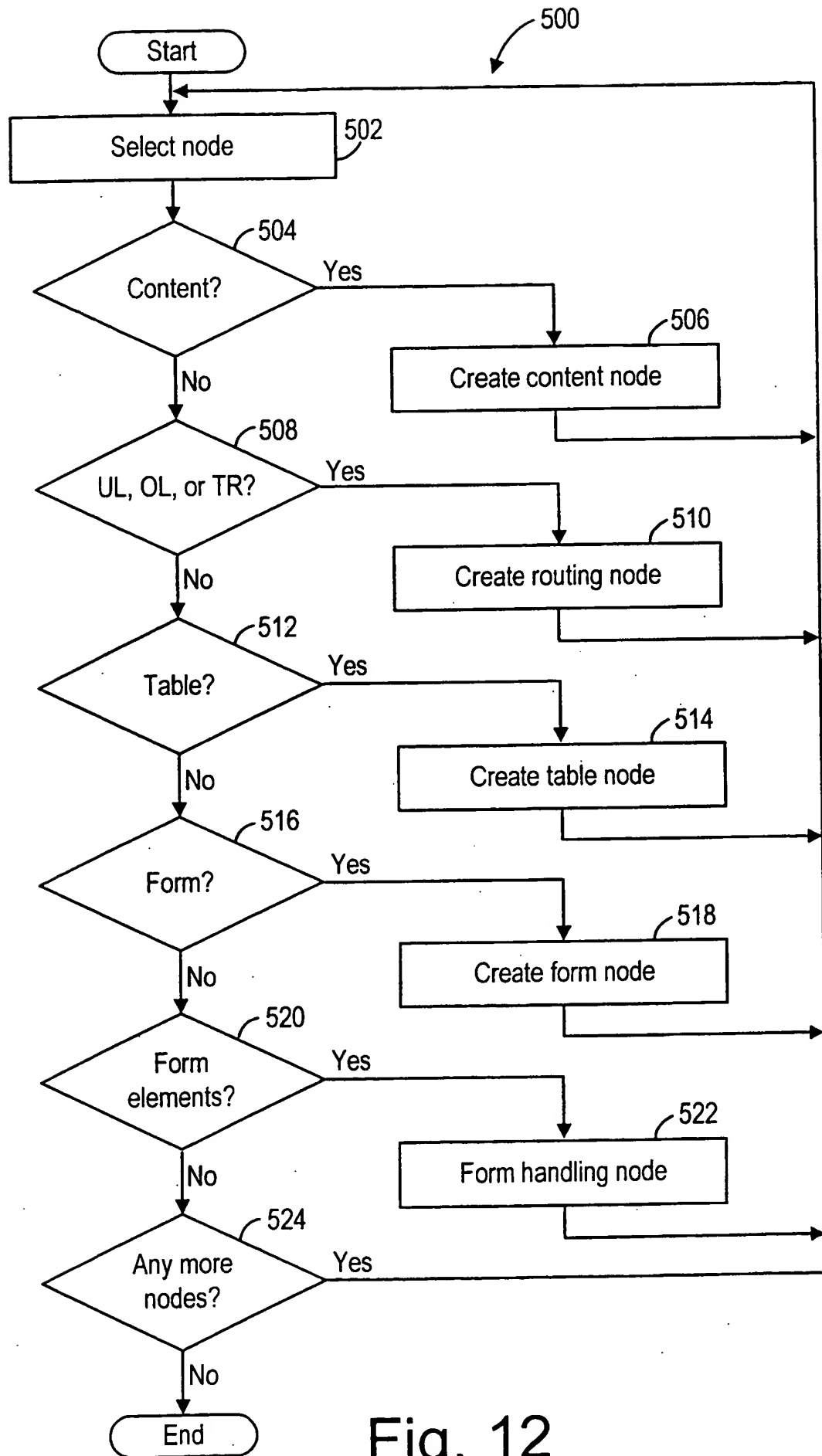


Fig. 12

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US00/30749**A. CLASSIFICATION OF SUBJECT MATTER**

IPC(7) : G06F 13/38, 15/17

US CL : 709/245, 379/88.18

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/245, 246, 203, 217; 379/88.18

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)  
DIALOG, EAST BRS**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,915,001 A (UPPALURU) 22 June 1999, col.4, 5, 7 10-12.	1, 2, 4-7, 12-15
Y		3, 8-11, 16-36
Y	US 5,899,975 A (NIELSEN) 04 May 1999, col. 1 lines 15-22; col. 9 lines 49-66; figure 9	8-10, 16-19, 22-36
Y	US 5,926,180 A (SHIMAMURA) 20 July 1999, throughout invention.	11, 20-21
Y	US 5,875,242 A (GLASER et al.) 23 February 1999, col. 1 lines 30-50.	3

☐ Further documents are listed in the continuation of Box C.
 ☐ See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

05 FEBRUARY 2001

Date of mailing of the international search report

01 MAR 2001

Name and mailing address of the ISA/US  
Commissioner of Patents and Trademarks  
Box PCT  
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

BUNJOB JAROEJONWANT

Telephone No. (703) 305-9731